EVOLVING TOGETHER, FAILING APART: THE SKILL–POLICY CO-EVOLUTION TRAP IN SELF-IMPROVING AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Skill–policy co-evolution—in which an agent's policy model and its skill bank improve each other during training—is a promising paradigm for self-improving agents. Current evaluations score the co-evolved policy on the final bank it trained with (the *delivered bank*). After deployment, however, skills may be rewritten, added or removed more frequently than the policy model is retrained, leaving the policy to operate on a drifted skill bank that neither the training objective nor existing evaluation protocols address. In this paper, we first establish, through theoretical and empirical analysis, that the fragility hidden by this blind spot is not incidental. The acceptance rule these methods share (*cooperative acceptance*) ties a policy's competence to the bank it co-evolved with, leaving it fragile once skills drift, regardless of the implementation; empirically, sensitivity to skill drift grows as co-evolution training proceeds. We therefore introduce the **CO**-evolution **D**rift **A**udit (**CODA**), a deployment-oriented evaluation protocol that simulates skill drift and, alongside success rate under worst-case drift, introduces two new metrics: a *consistency rate* under meaning-preserving edits and a *resistance rate* to corrupted skills. Across four co-evolution methods and five benchmarks (six settings), worst-case success is **5–13** points below the peak measured for each delivered pair, erasing **52–93%** of the delivered bank's gain over the no-skill baseline. CODA thus reframes the target of co-evolution—from peak performance on the co-evolved bank to the ability to harness skills however they evolve—and provides a shared standard against which future methods can be audited[1].

1 INTRODUCTION

Building agents that improve themselves is a central goal of current agent research, and co-evolving a skill library with a policy is one of the most promising routes (Xia et al., 2026a; He et al., 2026). The agent solves tasks, distills the experience into reusable skills, and uses those skills to strengthen its policy in turn, so competence compounds over rounds. A recent line of methods (Shi et al., 2026; Li et al., 2026; Zhang et al., 2026b; Yang et al., 2026a; Lin et al., 2026) follows exactly this recipe: it iterates the policy and the skill bank together during training and reports large gains on agent benchmarks. Each ultimately delivers a pair—a co-evolved policy and its final skill bank (the *delivered bank*)—and reports the policy's task success on that bank.

However, current evaluation is disconnected from deployment: it assumes that the skill bank remains fixed after training. Real skill banks are non-stationary: skills are rewritten, entries are added or removed, and versions coexist, often changing faster than the policy is retrained. A deployed policy therefore typically runs on a bank that has drifted from the one it was delivered with, yet existing protocols do not test whether performance is maintained. A few works note this brittleness (Liu et al., 2026d; Hebbar et al., 2026) but offer neither a systematic measurement nor a causal account.

We argue that the blind spot conceals a property of the training objective, not an incidental evaluation artifact. Co-evolution repeatedly adapts the policy and bank into a mutually specialized pair, while evaluation rewards exactly that pairing. The policy is therefore optimized for its delivered bank

[1] Source code is available at `https://anonymous.4open.science/r/CODA-EEC6/`.

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.

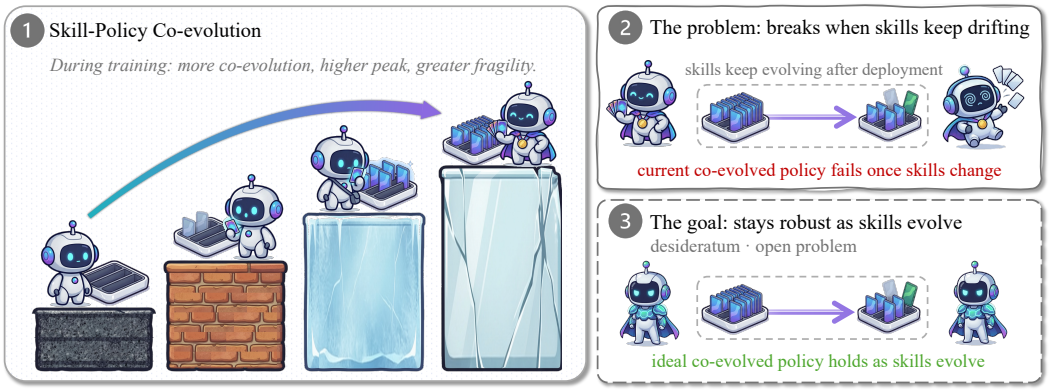


Figure 1: **The skill–policy co-evolution trap (conceptual illustration)**. **(1) Training**: The policy and skill bank co-evolve round by round, and evaluation reports the peak reached by the resulting delivered pair. **(2) Deployment**: The skill bank continues to evolve while the policy remains fixed. Because the policy is specialized to its delivered bank, its performance degrades as the bank drifts. **(3) Goal**: An ideally co-evolved policy that retains its performance as skills evolve.

rather than for transferable skill use. We establish this in two ways. Theoretically, we prove that the shared acceptance rule—retaining a skill update only when it helps the current policy—places the delivered pair at a peak over candidate banks, so rewriting the delivered skills cannot improve the policy and usually degrades its performance (Section 2). Empirically, using the proposed *co-evolution fragility curve*, we show that sensitivity to skill drift increases as training proceeds in five of six settings (Section 3). We call this the *co-evolution trap* (Figure 1): the process that raises peak performance also creates sensitivity to bank drift.

Establishing the problem exposes a second gap: existing evaluations cannot measure it, because they report only one success rate on the fixed delivered bank. We therefore introduce the **CO**-evolution **D**rift **A**udit (**CODA**), a deployment-oriented protocol for auditing post-deployment skill drift under controlled conditions. CODA supplements the conventional peak success rate with worst-case success and two new metrics—a *consistency rate* under meaning-preserving edits and a *resistance rate* to corrupted skills. Across four methods and five benchmarks (six settings), worst-case success is **5–13 points** below the peak measured for each delivered pair, erasing **52–93%** of the delivered bank's gain over the no-skill baseline, so the reported peak overstates the competence a deployment can rely on. CODA thus reframes the target of co-evolution—from achieving peak performance on the bank a method co-evolved with to using skills reliably as they keep changing—and we release it as a common standard for auditing future methods.

Our main contributions are summarized as follows:

- We **expose a trap in skill–policy co-evolution**: current evaluations measure fit to a fixed delivered bank, leaving performance under deployment-time bank change unmeasured.
- We **theoretically and empirically demonstrate** its mechanism and signature: under cooperative acceptance and candidate coverage, no covered rewrite improves the delivered policy; empirically, 18/18 endpoint pairs obey this prediction and fragility grows along training in five of six settings.
- We **propose a deployment-oriented evaluation protocol**, CODA, reporting worst-case success under tested content-level drift, consistency under meaning-preserving edits, and resistance to corrupted skills, making the deployment robustness of the delivered policy–bank pair measurable.

2 Why Co-evolution Favors Specialization over Robustness

Problem setting. A task x is drawn from a distribution D, a skill bank S lies in a space $\mathcal{S}$, and a policy $\pi_\theta(\cdot \mid x, S)$ maps both to a distribution over action sequences, with skills entering as text in its context. Methods optimize and report the expected return

$$J(\theta, S) \;=\; \mathbb{E}_{x\sim D}[R(\pi_\theta(\cdot \mid x, S))]\,, \tag{1}$$

Table 1: Current skill–policy co-evolution methods differ in implementation but share a selection bias toward current-policy utility and report peak return on the co-evolved bank.

	Design choices (vary across methods)				**Shared structure**	
Method	Cadence	Growth	Gate	Injection	Selection bias	Objective
SkillRL (Xia et al., 2026a)	alternating	monotone	success-gated	batch+top-k	**current utility**	**peak return**
ReSkill (He et al., 2026)	near-sync	capped ($\leq$8)	Thompson samp.	per-step		
SAPO (Zhang et al., 2026b)	near-sync	two-tier, capped	marginal util. >0	rerank		
Skill1 (Shi et al., 2026)	joint	monotone	success (r=1)	log-prob		
ARISE (Li et al., 2026)	joint	two-tier	hier. reward	log-prob		
SkillMaster (Yang et al., 2026a)	near-sync	managed	policy scoring	policy-managed		
Evolving-RL (Fan et al., 2026)	joint	monotone	reward thresh.	batch		
COSPLAY (Wu et al., 2026)	near-sync	managed (lifecycle)	contract verif.	per-step		

where R is a binary success indicator, so J is the reported success rate. We fold skill selection into π_θ, making J a function of the bank as a whole. Co-evolution alternates a *policy step*, which improves θ against the current bank, and a *bank step*, which proposes banks by rewriting, adding or removing skills and retains one through an acceptance test. After T rounds, it delivers (θ^*, S^*) and reports $J(\theta^*, S^*)$ as its *peak return*—the de facto objective for comparing these methods.

2.1 A Unified View and the Premise It Reveals

Current methods differ in *update cadence*, *bank growth*, *validation gate* and *injection mechanism* (Table 1; definitions in Appendix C). Across these choices, they share two patterns. First, they optimize and report peak return on the bank they co-evolved with. Second, their gates favor banks that serve the current policy, whether through reward thresholds, sampled-return comparisons or posteriors over competing versions. The first motivates the point-mass diagnosis of Section 2.3; the second motivates the idealized selection rule below.

No method searches all of $\mathcal{S}$. Let $\mathcal{C}_t \subseteq \mathcal{S}$ denote the candidate banks available at round t.

Assumption 1 (Idealized cooperative acceptance)**.** The retained bank maximizes the current policy's return over the round's candidate set,

$$S_t \;\in\; \arg\max_{S \in \mathcal{C}_t} J(\theta_t, S). \tag{2}$$

The rule is *cooperative* because selection serves the current policy rather than challenging it. Actual gates admit candidates that pass a non-decreasing utility test and terminate after a finite budget; they need not solve Eq. (2) exactly. We use the equation as an explicit idealization of their shared selection bias, restricted to the round's candidate set rather than all of $\mathcal{S}$.

2.2 Perturbation Response and the Co-evolution Peak

Definition 1 (Content-level neighborhood)**.** For a bank S, let $\mathcal{B}(S)$ denote a specified family of banks obtained by rewriting the textual content of skills already present while fixing the skill set and retrieval-side fields. We include S itself and exclude the empty bank.

Assumption 2 (Candidate coverage at delivery)**.** The final candidate family covers the content-level neighborhood of the delivered bank,

$$\mathcal{B}(S^*) \;\subseteq\; \mathcal{C}_T. \tag{3}$$

Assumption 2 makes explicit that the theory applies only to the covered rewrite family, not to arbitrary post-deployment edits. The empty bank is excluded because removal is an autonomy anchor rather than a content perturbation (Section 4.2). Fixing retrieval-side fields holds the selected skill set constant, as enforced experimentally (Section 3.1; Appendix E.5).

Definition 2 (Perturbation response and fragility)**.** The *response set* of a policy at a bank collects the return changes induced by content-level perturbation, $\mathcal{R}(\theta, S) = \{J(\theta, S') - J(\theta, S) : S' \in \mathcal{B}(S)\}$, and *fragility* is its worst case in magnitude,

$$\Phi(\theta, S) \;=\; -\min \mathcal{R}(\theta, S) \;=\; J(\theta, S) - \min_{S' \in \mathcal{B}(S)} J(\theta, S'). \tag{4}$$

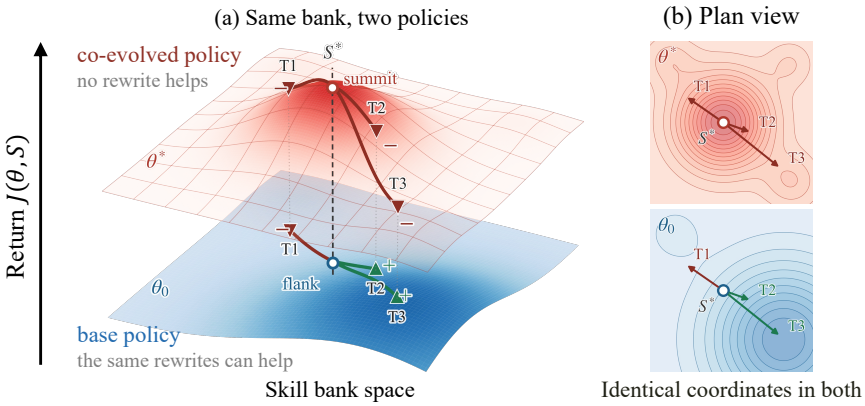


Figure 2: Proposition 1 as a return landscape (schematic). For the co-evolved policy θ^*, the delivered bank is a peak over the covered family, so no covered content perturbation ascends. A base policy θ_0 need not share that peak, and the same perturbations may improve its return.

The content-level drops measured in Section 3 (axes T1–T3) and the peak-to-worst gap reported in Section 4.2 are empirical lower bounds on Φ: both evaluate finitely many variants from $\mathcal{B}(S)$, so they can only understate its worst-case value, making our estimates conservative. Since $S \in \mathcal{B}(S)$, $\Phi \geq 0$ for *every* policy, so magnitude alone cannot distinguish a co-evolved policy from any other; what does is the *sign structure* of $\mathcal{R}$, which Proposition 1 makes precise.

Proposition 1 (Co-evolution peak)**.** *Under Assumptions 1 and 2, the delivered pair* $(\theta^*, S^*) = (\theta_T, S_T)$ *satisfies*

$$\mathcal{R}(\theta^*, S^*) \subseteq (-\infty, 0]. \tag{5}$$

Thus no covered content-level variant improves the co-evolved policy, and $\Phi(\theta^*, S^*) = 0$ *only if return is constant on* $\mathcal{B}(S^*)$*. For a policy* θ_0 *not selected against* S^* *under these assumptions, positive responses remain admissible:* $\max \mathcal{R}(\theta_0, S^*) > 0$ *may hold.*

Intuitively, selection places the delivered bank at a peak for the policy it co-evolved with, so no covered direction leads uphill (Figure 2). The result is one-sided: a policy not selected against this bank need not lie at the same peak, so a rewrite may help. Because both assumptions idealize the training process and its coverage, we test their prediction rather than claim to verify the premises: at delivery, no tested content-level perturbation improves performance beyond evaluation noise (Section 3).

2.3 The Cause: An Objective That Never Marginalizes over Banks

Proposition 2 (The point-mass objective prices degradation at zero)**.** *For any policy* θ *and distribution* P *with* $\mathrm{supp}(P) \subseteq \mathcal{B}(S^*)$*,*

$$\mathbb{E}_{S\sim P}[J(\theta, S)] \;=\; \underbrace{J(\theta, S^*)}_{\text{peak}} \;-\; \underbrace{\mathbb{E}_{S\sim P}[J(\theta, S^*) - J(\theta, S)]}_{\text{average degradation } \leq \Phi(\theta, S^*)}. \tag{6}$$

For the delivered policy θ^* *under Assumptions 1 and 2, the second term lies in* $[0, \Phi(\theta^*, S^*)]$*. The current point-mass case* $P = \delta_{S^*}$ *removes it by construction.*

Equation (6) is an algebraic identity and requires no hypothesis on the policy class. Under a richness condition, Appendix D.4 shows that equal peak return can coexist with different fragility, so optimizing the peak alone need not identify a robust policy. The issue is therefore not that co-evolution optimizes in the wrong direction, but that it optimizes at one bank while deployment presents a neighborhood. How to choose and sample from P without sacrificing peak return remains open.

Scope. Neither proposition assumes a particular model class, task distribution or reward; both apply to any method whose delivered pair satisfies Assumptions 1 and 2 (proofs in Appendix D). Under analogous coverage, reversing the selection preference drives Φ down instead (Appendix D.5), so the selection bias—not co-evolution alone—governs whether robustness emerges. The magnitude and training-time evolution of Φ are examined empirically in Section 3.2.

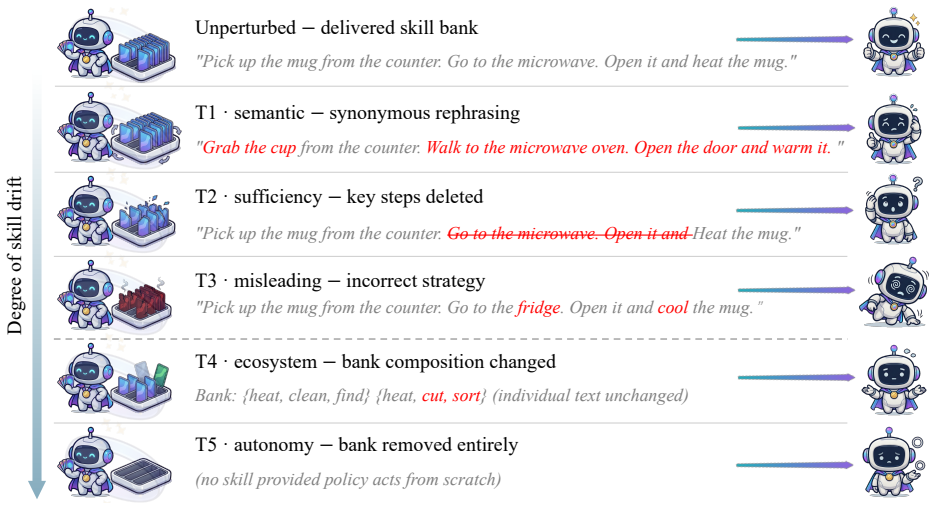


Figure 3: The five-axis probe on one ALFWorld task. Each row applies a single operator to the delivered bank and reports its effect on the same skill text, with the edit marked in red. T1–T3 alter selected skill content under fixed retrieval and lie within $\mathcal{B}(S^*)$; T4 changes bank composition, and T5 removes the bank. The task and policy remain fixed, and outcomes are schematic.

3 Empirical Evidence: Fragility Grows with Co-evolution

3.1 A Five-Axis Skill-Drift Probe

To measure the response set of Definition 2, we perturb the delivered skill bank while holding the task and policy fixed. Five axes represent increasingly structural forms of drift. **T1 (semantic)** rewrites a skill synonymously, changing wording but preserving meaning. **T2 (sufficiency)** deletes necessary steps or injects irrelevant content, leaving the skill on topic but incomplete. **T3 (misleading)** replaces the prescribed strategy with an incorrect one. **T4 (ecosystem)** changes bank composition by adding, removing or mixing skill versions. **T5 (autonomy)** removes the bank entirely. T1–T3 alter existing skill content and lie within $\mathcal{B}(S^*)$; T4 and T5 are reported as structural diagnostics outside it. Figure 3 illustrates all five axes; Appendix F provides detailed examples.

An external language model produces the variants from domain- and axis-specific prompts. For T1–T3, retrieval runs once on the original bank and only the selected content fields are replaced, preventing retrieval changes from confounding content changes. T4 and T5 alter bank composition by design. Full prompts, parameters, examples and masking details are in Appendices E, F and E.5.

3.2 Tracing Fragility Along Co-evolution

A two-point comparison between the base and delivered policies is difficult to compare across settings because base performance ranges from near zero to substantial. We instead trace checkpoints along one training trajectory while holding the delivered bank S^* and its perturbed variants fixed. The policy checkpoint is therefore the only changing component. This design yields our proposed *co-evolution fragility curve*, a within-run longitudinal diagnostic of whether fragility grows with co-evolution. Where available, the base (`s0`) checkpoint anchors the left endpoint; because it did not co-evolve with S^*, positive responses remain admissible under Proposition 1.

We retain checkpoints from the start of the run through its peak unperturbed success rate. Later checkpoints are not larger doses of *successful* co-evolution, whereas local training dips within the retained segment remain included (Appendix H).

Absolute and normalized readings. We track each axis with two numbers. The absolute *drop* (Drop; larger means more fragile) is the performance lost at a given checkpoint,

$$\mathrm{Drop}_a(t) \;=\; \mathrm{SR}(S^*) - \mathrm{SR}(S^*_a), \tag{7}$$

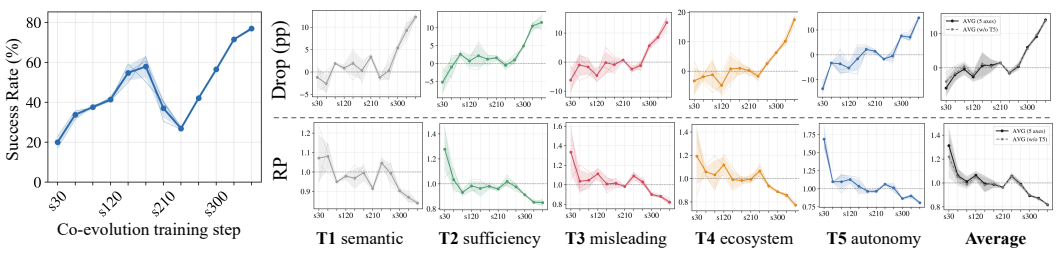


Figure 4: Co-evolution fragility curve for RESKILL on ALFWorld. **Left:** unperturbed success rate over the analyzed (rising) segment of training. **Right, top row:** absolute drop per axis; **right, bottom row:** retention performance RP per axis. Competence keeps rising (left) while the policy grows more sensitive to perturbation of its own bank—Drop trends upward and its normalized counterpart RP downward—so co-evolution buys peak performance at the cost of fragility.

where $\mathrm{SR}(S^*)$ is the unperturbed success rate of checkpoint t and $\mathrm{SR}(S_a^*)$ is its success rate under the axis-a perturbation. The *retention performance* (RP; higher is better) is competence-normalized,

$$\mathrm{RP}_a(t) \;=\; \frac{\mathrm{SR}(S_a^*)}{\mathrm{SR}(S^*)}, \tag{8}$$

so checkpoints of different ability compare on one scale (retaining 90% of the score gives $\mathrm{RP} = 0.9$ whether the baseline is 30% or 80%). Reporting both rules out the objection that a larger drop merely reflects more headroom, which retention normalizes away.

A minimal linear–Gaussian construction shows analytically that rising competence and fragility can emerge from the same co-evolution process (Appendix D.3).

3.3 SETUP AND RESULTS

Experimental setup. We evaluate four skill–policy co-evolution methods over six method–environment settings: SKILLRL (Xia et al., 2026a) on ALFWorld (Shridhar et al., 2020) and WebShop (Yao et al., 2022); RESKILL (He et al., 2026) on ScienceWorld (Wang et al., 2022) and ALFWorld; EVOLVING-RL (Fan et al., 2026) on Mind2Web (Deng et al., 2023); and ARISE (Li et al., 2026) on DeepScaleR-AMC23 (Luo et al., 2025). They share only the selection bias abstracted by Assumption 1, so a trend common to all four tracks the paradigm rather than one implementation. Appendix G details method- and benchmark-specific configurations.

Training trend. The result is consistent across four methods and five benchmarks: **as co-evolution training proceeds, the policy grows more capable with its own skill bank but increasingly sensitive to changes in it** (Figure 4). Unperturbed success rises from the first to the final retained anchor in all six settings, including $5.2\% \to 88.8\%$ for SKILLRL-ALFWorld, $16.4\% \to 25.2\%$ for EVOLVING-RL-Mind2Web and $15.5\% \to 87.4\%$ for ARISE-AMC23. At least one content-level axis has a significantly positive Drop slope in five of six settings; across the eighteen axis–setting pairs, twelve are significant—four on each of T1, T2 and T3. The effect is therefore not confined to misleading skills: semantic and sufficiency perturbations also become costlier in multiple settings.

The final delivered checkpoint is fragile in three concrete ways. (i) *The one-sided prediction holds (18/18):* no content-level rewrite beats evaluation noise in any of the eighteen axis–setting pairs, as Proposition 1 predicts. (ii) *The losses are substantial:* a meaning-preserving rewrite alone costs 10.9 points on RESKILL-ScienceWorld (peak 50.3%). (iii) *The bank still helps under drift:* $\mathrm{SR}_{\min}$ stays above the no-skill anchor in all six settings, so the failure is lost bank-derived gains, not negative skill value. Full results are reported in Appendix H: per-anchor grids in Tables 8–13, heat maps in Figures 5–10, and the remaining fragility curves in Figures 11–15.

The within-run trend and these delivered-pair results show one specialization pattern across distinct methods and domains; the next section turns this dependence into CODA, a checkpoint-free audit of any delivered policy–bank pair.

4 CODA: AUDITING CO-EVOLVED POLICIES FOR SKILL DRIFT

The previous two sections established, theoretically and empirically, that co-evolution leaves a policy fragile once its skills drift. What the field lacks, however, is not another method—new update rules or acceptance tests—but a way to *measure* the failure: a shared standard that scores how well a delivered policy holds up when its bank changes, comparably across methods. We therefore introduce CODA, a protocol that audits the one artifact every method ships—a co-evolved policy and its delivered bank—and reports its robustness to skill drift on a common footing.

4.1 DESIGN AND THE SHARED REWRITING SUITE

CODA audits only the delivered pair—a co-evolved policy and its delivered bank—plus the shared suite, and assigns scores only to quantities with a predefined direction, so the metric set stays closed; the design rationale (checkpoint-freeness, the direction-first rule, and why diagnostics such as the surface-form ratio κ of Appendix H.5 are excluded) is in Appendix E.

The suite is the part of this paper meant to outlive it, under one hard rule: **rewriting is performed by the released suite, not by the party under evaluation**—an evaluation whose conditions the evaluated party controls is not an evaluation. For the content-level axes T1–T3, retrieval is run once on the unperturbed bank and only the content of the selected skills is rewritten; the suite pins the rewriting model and its sampling, releases the perturbed banks rather than a script, preserves bank structure, and reports the edit strength it produced. The five requirements (R1–R5) that make this workable, with their full rationale, are in Appendix E.2.

4.2 A GATE AND THREE ROBUSTNESS METRICS

The protocol scores only T1–T3, the content-level axes within $\mathcal{B}(S^*)$ covered by Proposition 1. T4 (ecosystem) and T5 (autonomy) are reported but not scored because they lack a consistent operator or direction across methods. The no-skill condition anchors the gate and resistance criterion rather than constituting a score (Appendix E.8).

Fix a delivered pair (θ^*, S^*) and a test set of N tasks; let $c_i(S) \in \{0, 1\}$ record whether task i succeeds under bank S and $\mathrm{SR}(S) = \frac{1}{N}\sum_i c_i(S)$. Write $\varnothing$ for the empty bank and S^*_a for axis $a \in \{\mathrm{T1}, \mathrm{T2}, \mathrm{T3}\}$ applied to S^*. All comparisons are *paired*: the same tasks, policy and selected skills. A net-benefit gate admits the method, and three metrics then score it, one per content axis; each is defined with its pass line below, and the supporting derivations (the score-invariance bound and the fragility lower bound) are collected in Appendix E.8.

Gate: net skill benefit. A bank must earn its place before its robustness matters,

$$\Delta_{\mathrm{net}} \;=\; \mathrm{SR}(S^*) - \mathrm{SR}(\varnothing). \tag{9}$$

A method with $\Delta_{\mathrm{net}} \le 0$ does not qualify. A small positive Δ_{net}, below the noise band, is admitted but flagged: it leaves no headroom to tell degradation from noise.

Metric 1: consistency rate (CR), higher is better. A meaning-preserving rewrite (T1) should not alter task outcomes. The CR measures the fraction of outcomes unchanged by the rewrite,

$$\mathrm{CR} \;=\; \frac{1}{N}\left|\{\, i : c_i(S^*_{\mathrm{T1}}) = c_i(S^*) \,\}\right|, \tag{10}$$

where $\mathrm{CR} = 1$ is ideal. Unlike $|\Delta\mathrm{SR}|$, CR captures opposing outcome flips that cancel at the aggregate level. To account for run-to-run variation, we use the consistency CR_0 between two unperturbed runs as the baseline and declare a pass when $\mathrm{CR} \ge \mathrm{CR}_0 - 2\sigma_0$ (Appendix E.8).

Metric 2: resistance rate (RR), higher is better. A judicious policy should resist an incomplete (T2) or incorrect (T3) skill by falling back on its own competence. We define the *resistance rate* as the fraction of the full test set solved both without skills and under corruption:

$$\mathrm{RR}_a \;=\; \frac{1}{N}\left|\{\, i : c_i(\varnothing) = 1 \wedge c_i(S^*_a) = 1 \,\}\right|. \tag{11}$$

Thus, RR_a is a within-setting diagnostic rather than a cross-benchmark ranking. Pass/fail is determined separately by the harmful-axis criterion

$$\mathrm{SR}(S_a^*) \;\geq\; \mathrm{SR}(\varnothing), \tag{12}$$

evaluated on paired tasks using McNemar's test. We report T2 and T3 separately because this criterion is stringent for incorrect skills but lenient for incomplete ones that retain partial value.

Metric 3: worst-case success rate ($\mathrm{SR}_{\min}$), higher is better. The worst-case success rate is the lowest mean performance across T1–T3:

$$\mathrm{SR}_{\min} = \min_{a\in\{\mathrm{T1,T2,T3}\}} \overline{\mathrm{SR}}(S_a^*), \qquad \overline{\mathrm{SR}}_{\mathrm{avg}} = \frac{1}{3}\sum_a \overline{\mathrm{SR}}(S_a^*), \tag{13}$$

where the bar denotes the mean over repeats and $\overline{\mathrm{SR}}_{\mathrm{avg}}$ is reported secondarily. The gap $\mathrm{SR}(S^*) - \mathrm{SR}_{\min}$ lower-bounds the fragility Φ in Definition 2; Appendix E.8 provides the derivation.

4.3 Verdict: An Absolute, Within-Method Pass or Fail

CODA's verdict is an absolute pass or fail, judged within a single method: it passes when the gate clears ($\Delta_{\mathrm{net}} > 0$), the consistency rate stays within the CR_0 noise band, and every harmful axis satisfies Eq. (12). Every check reads a method against its own numbers, never against another's; our headline—that the reported peak conceals a much lower worst case—therefore ranks nothing and needs no cross-method comparison. Ranking is a use we leave to adopters: once some pass, $\mathrm{SR}_{\min}$ is the natural leaderboard key, and three raw quantities are published regardless—the peak, the no-skill rate, and the per-axis perturbed rates.

Auditing each method on the bank it produced is a deliberate choice, not a concession: a robustness protocol standardizes the operator and its budget, not the perturbation text, and no shared bank exists in deployment anyway. The full argument—including why a bank too vague to corrupt is caught by the gate—is in Appendix E.9.

4.4 Main Result: the Report Card

We audit four co-evolution methods across the six method–environment settings of Section 3.3, running CODA on each final delivered pair—not any intermediate checkpoint—with perturbed banks from the shared rewriting suite. To keep the comparison fair, the peak we report is our own re-evaluation of each pair under a single fixed protocol, not the number in the original paper; because backbones differ, absolute numbers compare "method + backbone" rather than the method alone (Appendix G). Table 2 is the audit: the top block gives the headline gap between the peak and the worst case once the bank drifts, and every other row is what CODA adds—how that same delivered system behaves once it no longer controls the wording of its bank.

The reported peak overstates deployable competence. Across all six settings the worst-case success rate under content-level drift trails the peak by 4.9–12.6 points (about 5–13): 4.9 on EVOLVING-RL-Mind2Web, 5.0 on SKILLRL-WebShop, 6.5 on ARISE-AMC23 and 10.2–12.6 on the other three. Only the peak is customarily reported. Because the absolute drop scales with how much the bank helps, we normalize by that headroom: drift consumes 52–93% of the skill benefit in every setting. The worst-case rate nonetheless stays above the no-skill baseline throughout, so the failure is not that skills hurt but that most of the advertised gain is contingent on the bank's exact wording and disappears once it drifts. What a deployment can rely on is therefore nearer this worst case than the peak, by the majority of the benefit skills appear to add.

The worst case is often the mildest change. In three of the six settings the T1 (meaning-preserving) rate falls *below* the harsher T2 and T3 rates (e.g. SKILLRL-ALFWorld 78.6 vs. 79.3/79.5), costing up to 12.6 points relative to the peak (RESKILL-ALFWorld 76.9→64.3)—evidence that the policy has overfit to the bank's phrasing rather than its meaning. The consistency rate makes the churn explicit: at 77.5–87.0 it sits below its noise floor $\mathrm{CR}_0 = 91.9$–94.4, so 13–23% of tasks flip outcome under a meaning-preserving rewrite, against 6–8% under decoding noise alone; the consistency criterion therefore fails in every setting—and with it the verdict—while the

Table 2: CODA report card. The top row is the peak these methods report, re-evaluated under one fixed protocol; the rows below show how much of it survives once the delivered bank drifts. The headline drop (shaded) is 5–13 points and erases 52–93% of the skill benefit over the no-skill baseline, and the consistency rate stays below its noise floor CR_0 throughout—so the reported peak overstates the competence that actually deploys. Details in Appendix H.

	SkillRL		ReSkill		Evolving-RL	ARISE
Metric	ALFWorld	WebShop	ScienceWorld	ALFWorld	Mind2Web	AMC23
peak success rate $\mathrm{SR}(S^*)$	88.8	52.4	50.3	76.9	25.2	87.4
worst case $\mathrm{SR}_{\min}$	78.6	47.4	39.4	64.3	20.3	80.9
peak→worst drop	**10.2**	**5.0**	**10.9**	**12.6**	**4.9**	**6.5**
Per-axis success rate						
T1 reword	78.6	51.5	39.4	64.3	23.2	82.5
T2 incomplete	79.3	47.4	47.7	66.6	20.3	81.5
T3 misleading	79.5	51.7	47.7	65.4	20.4	80.9
no-skill anchor $\mathrm{SR}(\varnothing)$	74.5	45.9	34.2	63.3	15.8	75.3
Robustness metrics (higher better; not success rates)						
consistency rate CR	87.0	83.2	86.1	84.2	77.5	82.1
noise floor CR_0	92.9	91.9	93.6	92.2	94.4	93.7
resistance rate $\mathrm{RR}_{\mathrm{T2}}$	70.2	37.8	31.0	60.6	12.2	63.5
resistance rate $\mathrm{RR}_{\mathrm{T3}}$	71.2	40.0	33.5	58.9	11.4	65.4

net-benefit gate and the harmful-axis criterion pass. CODA does not declare these methods useless: it quantifies how much of each reported peak is co-evolved with the delivered bank, precisely the quantity current evaluation leaves unmeasured.

5 Related Work

Skill–policy co-evolution. A growing family of methods co-evolves a skill library with the policy, including SkillRL (Xia et al., 2026a), ReSkill (He et al., 2026), Skill1 (Shi et al., 2026), ARISE (Li et al., 2026), SAPO (Zhang et al., 2026b), and SkillMaster (Yang et al., 2026a). These methods differ in update cadence, library growth, validation gates, and skill injection (Table 1), yet share the same evaluation premise: the co-evolved policy is tested on its delivered bank. We introduce no new member of this family; instead, we examine this shared premise and its consequences under deployment-time skill drift.

Agent robustness under skill drift. Robustness evaluations for language agents typically perturb the task side of the interaction—instructions, observations, tool outputs, or environment dynamics. We instead perturb the agent's own procedural knowledge, which task-side benchmarks treat as fixed. Recent work documents that skill libraries drift and that agent performance can degrade accordingly (Zhang et al., 2026a; Chen et al., 2026b; Liu et al., 2026b; Chen et al., 2026a), that skills may harm rather than help (Tank & Nama, 2026; Yu et al., 2026), and that self-evolution is not always faithful (Zhao et al., 2026; Lv et al., 2026). Other approaches repair skills post hoc or construct more drift-tolerant skills (Liu et al., 2026c; Park et al., 2026; Shao et al., 2026); by contrast, we audit the delivered policy's dependence on whichever bank it encounters. This setting also resembles zero-shot coordination (Hu et al., 2020), with the skill bank acting as a co-evolved partner that may change after training. Appendix B provides further discussion.

6 Conclusion

This work shifts the evaluation of skill–policy co-evolution from performance on a single delivered pair to robustness under realistic deployment-time skill drift. Our analysis identifies cooperative acceptance as a source of dependence on the co-evolved bank, while the *co-evolution fragility curve* shows that this dependence can strengthen over training. CODA operationalizes these findings by auditing whether a delivered policy remains consistent under benign rewrites and resists incomplete or misleading skills. The delivered peak is therefore informative but insufficient: future methods should preserve its gains while learning to interpret, reject, and adapt to skills as the bank evolves.

AI USE STATEMENT

Large language models play two distinct roles in this work. As a *component of the method*, the CODA rewriting suite calls `gpt-4o-2024-08-06` (temperature 0.7, with version and sampling parameters pinned as specified in Appendix E) to generate the perturbed skill banks that serve as evaluation stimuli; because these banks are released, reproducing our evaluation does not require re-running the model. As an *authoring aid*, LLMs assisted only with language polishing, grammar, and style. All research questions, methodology, experiments, analyses, and conclusions were conceived, implemented, and verified by the authors, who take full responsibility for the content of this paper.

ETHICS STATEMENT

The CODA suite deliberately produces incomplete (T2) and misleading (T3) procedural instructions in order to measure how a delivered policy withstands skill drift. These corrupted skills are generated and consumed entirely within the benchmark harness, are never deployed or shown to real users, and are released alongside the suite so that their content is transparent and auditable. The work involves no human participants, collects no personally identifiable information, and introduces no new sensitive data; we see no direct risks to privacy, fairness, or legal compliance, and the work complies with the ICLR Code of Ethics.

REPRODUCIBILITY STATEMENT

We take several concrete measures to make our results reproducible.

- **Rewriting suite and protocol:** the rewriting model, its version, and all sampling parameters are pinned (Section 4.1), and the domain contexts, layered prompts, and resulting perturbed banks are released, so the evaluation can be reproduced without re-running the generative model.
- **Environments and data:** all benchmarks used (ALFWorld, WebShop, ScienceWorld, Mind2Web, and DeepScaleR-AMC23) are publicly available; the test splits, episode counts, and evaluation configuration are documented in Section 4.4 and Appendix G, together with the released configuration.
- **Code and logs:** the source code, evaluation scripts, and per-episode outcome logs are available at `https://anonymous.4open.science/r/CODA-EEC6/`.

Together, these measures allow independent researchers to reproduce our results and verify the claims made in this paper.

REFERENCES

Charles Chen, Qiming Yu, Yuhang Gu, Zhuoye Huang, Hanjing Li, Hongyu Liu, Simin Liu, Jinhao Liu, Dengyun Peng, Jiangyi Wang, et al. The scaling laws of skills in llm agent systems. *arXiv preprint arXiv:2605.16508*, 2026a.

Jianming Chen, Xuanbin Ye, Yawen Wang, Junjie Wang, Qing Wang, and Fanjiang XU. Vce-skill: Enhancing skill self-evolution with version-change experience. *arXiv preprint arXiv:2608.16544*, 2026b.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36:28091–28114, 2023.

Zhiyuan Fan, Wenwei Jin, Feng Zhang, Bin Li, Yihong Dong, Yao Hu, and Jiawei Li. Evolving-rl: End-to-end optimization of experience-driven self-evolving capability within agents. *arXiv preprint arXiv:2605.10663*, 2026.

Jinyuan Feng, Dongmin Li, Yiqun Chen, Yang Gao, Xing Chen, Huimu Wang, and Zhiqiang Pu. Coskill: Joint reinforcement learning of reasoning and meta-skill agents for hierarchical skill evolution. *arXiv preprint arXiv:2609.04865*, 2026.

Huan-ang Gao, Jiayi Geng, Wenyue Hua, Mengkang Hu, Xinzhe Juan, Hongzhang Liu, Shilong Liu, Jiahao Qiu, Xuan Qi, Yiran Wu, et al. A survey of self-evolving agents: What, when, how, and where to evolve on the path to artificial super intelligence. *arXiv preprint arXiv:2507.21046*, 2025.

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

Prannay Hebbar, Yogendra Manawat, Samuel Verboomen, Alesia Ivanova, Selvam Palanimalai, Kunal Bhatia, and Vignesh Baskaran. Sia: Self improving ai with harness & weight updates. *arXiv preprint arXiv:2605.27276*, 2026.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

Hengyuan Hu, Adam Lerer, Alex Peysakhovich, and Jakob Foerster. "other-play" for zero-shot coordination. In *International conference on machine learning*, pp. 4399–4410. PMLR, 2020.

Siyuan Huang, Pengyu Cheng, Haotian Liu, Tao Chen, Yihao Liu, Jingwei Ni, Shijie Zhou, Ziyi Yang, Gangwei Jiang, Mengyu Zhou, et al. Skill self-play: Pushing the frontier of llm capability with co-evolving skills. *arXiv preprint arXiv:2607.22529*, 2026.

Xinyi Ke, Kai Li, Junliang Xing, Yifan Zhang, and Jian Cheng. Game-theoretic co-evolution for llm-based heuristic discovery. *arXiv preprint arXiv:2601.22896*, 2026.

Yu Li, Rui Miao, Zhengling Qi, and Tian Lan. Arise: Agent reasoning with intrinsic skill evolution in hierarchical reinforcement learning. *arXiv preprint arXiv:2603.16060*, 2026.

Huizu Lin, Chengkai Huang, Tianqi Gao, Tao Huang, Daijiao Liu, Tongxin Li, Xiaoyan Sun, and Lina Yao. Auso: Action-level unified skill optimization from internalization to utilization. *arXiv preprint arXiv:2608.21292*, 2026.

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. 2026a.

Yujian Liu, Jiabao Ji, Li An, Tommi Jaakkola, Yang Zhang, and Shiyu Chang. How well do agentic skills work in the wild: Benchmarking llm skill usage in realistic settings. *arXiv preprint arXiv:2604.04323*, 2026b.

Yuxuan Liu, Zhaochen Su, Lingyun Xie, Yuhao Zhang, Qing Zong, Jiahe Guo, Zhongwei Xie, Yiyan Ji, Yauwai Yim, Hongyu Luo, et al. Skillrevise: Improving llm-authored agent skills via trace-conditioned skill revision. *arXiv preprint arXiv:2606.01139*, 2026c.

Yuxuan Liu, Zhaochen Su, Yuhao Zhang, Jiahe Guo, Zhongwei Xie, Huihao Jing, Lingyun Xie, Qing Zong, Yauwai Yim, Zhixiong Zhang, et al. Rethinking self-evolving agent skills: Feedback dynamics over multiple rounds. *arXiv preprint arXiv:2608.02636*, 2026d.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Li Erran Li, Raluca Ada Popa, et al. Deepscaler: Surpassing o1-preview with a 1.5 b model by scaling rl. *Notion Blog*, 3(5), 2025.

Song-Lin Lv, Weiming Wu, Rui Zhu, Zi-Jian Cheng, and Lan-Zhe Guo. Can agents generalize to the open world? unveiling the fragility of static training in tool use. *arXiv preprint arXiv:2607.01084*, 2026.

Bo Mao, Jie Zhou, Yutao Yang, Xin Li, Xian Wei, Qin Chen, Xingjiao Wu, and Liang He. Learning while acting: A skill-enhanced test-time co-evolution framework for online lifelong learning agents. *arXiv preprint arXiv:2606.04815*, 2026.

Zhenyu Pan, Yiting Zhang, Zhuo Liu, Yolo Yunlong Tang, Zeliang Zhang, Haozheng Luo, Yuwei Han, Jianshu Zhang, Dennis Wu, Hong-Yu Chen, et al. Advevo-marl: Shaping internalized safety through adversarial co-evolution in multi-agent reinforcement learning. *arXiv preprint arXiv:2510.01586*, 2025.

Jongchan Park, Seungjun Oh, Seungho Baek, and Yusung Kim. Learning generalizable skill policy with data-efficient unsupervised rl. *arXiv preprint arXiv:2607.00392*, 2026.

Zhe Ren, Yimeng Chen, Dandan Guo, Guowei Rong, Tonghui Li, RB Xiong, Qingfeng Lan, Wenyi Wang, Li Nanbo, Yibo Yang, et al. Self-improvements in modern agentic systems: A survey. *arXiv preprint arXiv:2607.13104*, 2026.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of nlp models with checklist. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 4902–4912, 2020.

Jie-Jing Shao, Haiyan Yin, Yueming Lyu, Xingrui Yu, Lan-Zhe Guo, Ivor Tsang, James Kwok, and Yu-Feng Li. Lifting traces to logic: Programmatic skill induction with neuro-symbolic learning for long-horizon agentic tasks. *arXiv preprint arXiv:2605.01293*, 2026.

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

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*, 2020.

Darshan Tank and Baran Nama. The regression tax: Decomposing why skills help and hurt llm agents. *arXiv preprint arXiv:2607.22520*, 2026.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. Scienceworld: Is your agent smarter than a 5th grader? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11279–11298, 2022.

Xiaoyu Wen, Zhida He, Han Qi, Ziyu Wan, Zhongtian Ma, Ying Wen, Tianhang Zheng, Xingcheng Xu, Chaochao Lu, and Qiaosheng Zhang. Magic: A co-evolving attacker-defender adversarial game for robust llm safety. *arXiv preprint arXiv:2602.01539*, 2026.

Xiyang Wu, Zongxia Li, Guangyao Shi, Alexander Duffy, Tyler Marques, Matthew Lyle Olson, Tianyi Zhou, and Dinesh Manocha. Co-evolving llm decision and skill bank agents for long-horizon tasks. *arXiv preprint arXiv:2604.20987*, 2026.

Peng Xia, Jianwen Chen, Hanyang Wang, Jiaqi Liu, Kaide Zeng, Yu Wang, Siwei Han, Yiyang Zhou, Xujiang Zhao, Haifeng Chen, et al. Skillrl: Evolving agents via recursive skill-augmented reinforcement learning. *arXiv preprint arXiv:2602.08234*, 2026a.

Peng Xia, Jianwen Chen, Xinyu Yang, Haoqin Tu, Jiaqi Liu, Kaiwen Xiong, Siwei Han, Shi Qiu, Haonian Ji, Yuyin Zhou, et al. Metaclaw: Just talk–an agent that meta-learns and evolves in the wild. *arXiv preprint arXiv:2603.17187*, 2026b.

Min Yang, Jinghua Piao, Xu Xia, Xiaochong Lan, Jiaju Chen, Yongshun Gong, and Yong Li. Skillmaster: Toward autonomous skill mastery in llm agents. *arXiv preprint arXiv:2605.08693*, 2026a.

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

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

Jianxiang Yu, Jiapeng Zhu, Bochen Lin, Qier Cui, Zichen Ding, and Xiang Li. Skill is not one-size-fits-all: Model-aware skill alignment for llm agents. *arXiv preprint arXiv:2605.30723*, 2026.

Xing Zhang, Yanwei Cui, Guanghui Wang, Ziyuan Li, Wei Qiu, Bing Zhu, and Peiyang He. Library drift: Diagnosing and fixing a silent failure mode in self-evolving llm skill libraries. *arXiv preprint arXiv:2605.19576*, 2026a.

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

Weixiang Zhao, Yingshuo Wang, Yichen Zhang, Yang Deng, Yanyan Zhao, Wanxiang Che, Bing Qin, and Ting Liu. Large language model agents are not always faithful self-evolvers. *arXiv preprint arXiv:2601.22436*, 2026.

Yingli Zhou, Wang Shu, Yaodong Su, Wenchuan Du, Yixiang Fang, and Xuemin Lin. A comprehensive survey on agent skills: Taxonomy, techniques, and applications. *arXiv preprint arXiv:2605.07358*, 2026.

Henry Peng Zou, Wei-Chieh Huang, Yaozu Wu, Jizhou Guo, Yankai Chen, Chunyu Miao, Hoang Nguyen, Yue Zhou, Weizhi Zhang, Liancheng Fang, Hanrong Zhang, Fangxin Wang, Pengfei Zhang, Huacan Wang, Langzhou He, Yangning Li, Dongyuan Li, Renhe Jiang, Xue Liu, and Philip S. Yu. Llm-based human-agent collaboration and interaction systems: A survey, 2026. URL `https://arxiv.org/abs/2505.00753`.

A LIMITATIONS

We note three limitations of this work.

L1 *Rewriting is not bit-reproducible.* A language model rewriting a skill is not a deterministic operator: pinned model versions, pinned sampling, and released perturbed banks reduce the variance a re-runner faces but do not remove it. If the bank was written by the same model family that later rewrites it, the measured fragility may be optimistically low.

L2 *We diagnose but do not fix.* We expose the fragility and provide a measurement tool, but propose no mitigation. The natural direction is the distributional objective of Eq. (6)—training against a distribution of bank perturbations rather than a single fixed bank—but turning it into a practical method, and keeping the net-benefit gate satisfied, is future work.

L3 *Scale is unstudied.* Whether broader exposure—more benchmarks, larger banks, more data—reduces sensitivity to wording or merely raises the peak from which the worst case falls away is beyond our design, and is the most interesting open question beyond the fix itself.

B EXTENDED RELATED WORK

Skill–policy co-evolution. A growing family of methods lets an agent distill task experience into reusable skills and then trains the policy on the enlarged bank (Liu et al., 2026a; Zou et al., 2026; Yang et al., 2026b; Feng et al., 2026). SKILLRL (Xia et al., 2026a) distills successful trajectories into skills that are added to a monotonically growing library and gated by reward on sampled rollouts. RESKILL (He et al., 2026) caps the library at a fixed size and uses Thompson sampling to decide which incumbent a new candidate should replace. SKILL1 (Shi et al., 2026) jointly optimizes the policy and bank end-to-end, admitting skills from trajectories that already succeed. ARISE (Li et al., 2026) evolves a two-tier hierarchy of intrinsic skills under a hierarchical reward. SAPO (Zhang et al., 2026b) admits a skill only when its estimated marginal utility to the current policy is positive. SKILLMASTER (Yang et al., 2026a) lets the policy itself score candidates, retaining high-scoring ones into a managed library. These methods differ in cadence, library growth, validation gate, and injection mechanism (Table 1; Appendix C gives the full classification). What they do not differ in is how success is reported: the co-evolved policy is run on the delivered bank and a task success rate is given. Surveys chart this space (Zhou et al., 2026; Gao et al., 2025; Ren et al., 2026), and recent variants push co-evolution to test-time (Mao et al., 2026), self-play (Huang et al., 2026), and deployment-time meta-learning (Xia et al., 2026b). We contribute no new member of this family. Our object of study is the family itself—specifically, the one premise that survives all of their design disagreements.

Agent robustness and adversarial co-evolution. Work on robustness for language agents perturbs the task side of the interaction—instructions, observations, tool outputs, environment dynamics—and asks whether performance survives. We perturb a different thing: the agent's own store of procedural knowledge, which no task-side benchmark touches because it is normally treated as part of the agent rather than part of the environment. Concurrently, a cluster of works documents that skill libraries drift and that agents degrade under that drift (Zhang et al., 2026a; Chen et al., 2026b; Liu et al., 2026b; Chen et al., 2026a), that skills can hurt as often as they help (Tank & Nama, 2026; Yu et al., 2026), and that self-evolution is not always faithful (Zhao et al., 2026; Lv et al., 2026). These diagnose the symptom; none ties it to the co-evolution objective or measures it on the delivered artifact. SkillRevise (Liu et al., 2026c) repairs skills post hoc, and GenDa (Park et al., 2026) with programmatic skill induction (Shao et al., 2026) build drift-tolerant skills from the skill side; we instead audit the policy's dependence on whatever bank it is handed. The closest conceptual relative is zero-shot coordination (Hu et al., 2020), where a policy trained with one partner must act with another it never co-evolved with; our setting is the single-agent analog, with the skill library in the role of the partner that changed. A separate line evolves an adversary alongside the learner, in game-theoretic heuristic discovery (Ke et al., 2026) or attacker–defender safety training (Wen et al., 2026; Pan et al., 2025), and reports robustness *gains*. This is not a counterexample to Proposition 1 but a confirmation of its premise read in reverse: those methods accept a candidate precisely when it *lowers* the current policy's return, so the sign in Assumption 1 is flipped and the conclusion flips with it. Appendix D.5 works this through, and the comparison is instructive rather than competitive: it locates the fragility in the acceptance rule rather than in co-evolution as such.

Standardized perturbation suites. CODA belongs to a tradition of fixed perturbation suites applied to models their designers did not build: corruption benchmarks in vision (Hendrycks & Dietterich, 2019) and behavioral test suites in NLP (Ribeiro et al., 2020) both consist of a released operator, a reporting convention, and the finding that published systems fail more than their headline numbers suggest. We follow that template on the skill side. One difference must be stated rather than glossed over: those operators are deterministic algorithms that reproduce bit-for-bit, whereas rewriting a skill with a language model does not. We narrow the gap: we pin the rewriting model, its version and its sampling parameters; we ship the domain context so nothing is left to the user; and we release the perturbed banks alongside the suite (Section 4.1). We do not close it, though—Section 6 lists it first among the limitations.

C Classification of Existing Methods

The four axes of Table 1 are defined as follows. *Update cadence* determines how the two steps interleave, from strictly alternating through near-synchronous to jointly optimized. The *validation gate* determines how a candidate bank is admitted: no test at all, an improvement test on sampled rollouts, or a posterior test such as Thompson sampling. *Library growth* is either monotone, so that skills accumulate, or capped, so that admission displaces an incumbent. The *injection mechanism* determines what reaches the context at inference: a batch selected by task category, or a small subset triggered per step.

We now place each surveyed method on these axes and state its acceptance rule—the property that Assumption 1 abstracts. In every case the gate admits a candidate bank only when it does not lower (and usually strictly raises) the current policy's return, however differently that test is implemented.

- **SkillRL** (Xia et al., 2026a): alternating cadence, monotone growth, batch top-k injection. A distilled skill is admitted whenever it does not reduce reward on sampled rollouts and is never removed thereafter, so the retained bank is the reward-non-decreasing one.
- **ReSkill** (He et al., 2026): near-synchronous cadence, library capped at eight, per-step injection. A candidate is validated by Thompson sampling and kept only if its posterior return improves on the incumbent it would displace.
- **SAPO** (Zhang et al., 2026b): near-synchronous cadence, two-tier capped growth, reranked injection. A skill enters only when its estimated marginal utility to the current policy is positive.
- **Skill1** (Shi et al., 2026): joint optimization, monotone growth, log-probability injection. Skills are distilled from trajectories that already succeed ($r{=}1$) and trained jointly with the policy, so what is retained is exactly what supported a return the policy did not lose.

- **ARISE** (Li et al., 2026): joint optimization, two-tier growth, log-probability injection. Intrinsic skills are evolved under a hierarchical reward and retained on improvement of that reward.
- **SKILLMASTER** (Yang et al., 2026a): near-synchronous cadence, policy-managed growth and injection. The policy itself scores candidate skills and retains the high-scoring ones, i.e. those that raise its own return.
- **Evolving-RL** (Fan et al., 2026): joint optimization, monotone growth, batch injection. A skill is admitted end-to-end when it clears a reward threshold on the current policy.
- **COSPLAY** (Wu et al., 2026): near-synchronous cadence, lifecycle-managed growth, per-step injection. Skills pass a contract-verification step and are retained over their lifecycle while they continue to serve the policy's return.

The gates differ in form—an improvement test on rollouts, a Thompson posterior, a marginal-utility check, a reward threshold, a policy score, a contract—but all favor skills that serve the current policy. Assumption 1 idealizes this shared selection bias. Proposition 1 is model-agnostic but additionally requires the candidate-coverage condition of Assumption 2; it therefore applies conditionally rather than automatically to the eight surveyed methods.

D PROOFS AND THE MINIMAL LINEAR–GAUSSIAN CONSTRUCTION

D.1 SETUP AND NOTATION

We restate the objects used throughout. A task x is drawn from D, a skill bank S lives in $\mathcal{S}$, and the return of a policy is $J(\theta, S) = \mathbb{E}_{x\sim D}[R(\pi_\theta(\cdot \mid x, S))]$ as in Eq. (1). At round t the candidate set $\mathcal{C}_t \subseteq \mathcal{S}$ collects the banks the method's proposal mechanism could have produced and its gate could have retained; Assumption 1 states $S_t \in \arg\max_{S\in\mathcal{C}_t} J(\theta_t, S)$. The content-level neighborhood $\mathcal{B}(S)$ of Definition 1 contains a specified family of textual rewrites, includes S itself and excludes the empty bank. Assumption 2 requires $\mathcal{B}(S^*) \subseteq \mathcal{C}_T$ only at delivery. The response set $\mathcal{R}$ and fragility Φ are as in Definition 2. We write $(\theta^*, S^*) = (\theta_T, S_T)$ for the delivered pair and drop the round index when it is not needed.

D.2 PROOF OF PROPOSITION 1

Proof. For the first claim, Assumption 1 at round $t = T$ gives $J(\theta^*, S^*) = \max_{S\in\mathcal{C}_T} J(\theta^*, S)$, hence $J(\theta^*, S') \le J(\theta^*, S^*)$ for every $S' \in \mathcal{C}_T$. Since $\mathcal{B}(S^*) \subseteq \mathcal{C}_T$ by Assumption 2, the inequality holds in particular for every $S' \in \mathcal{B}(S^*)$, so each element $J(\theta^*, S') - J(\theta^*, S^*)$ of $\mathcal{R}(\theta^*, S^*)$ is non-positive, which is Eq. (5).

For the second claim, $\Phi(\theta^*, S^*) = 0$ means $\min_{S'\in\mathcal{B}(S^*)} J(\theta^*, S') = J(\theta^*, S^*)$. Combined with the upper bound just established, this forces $J(\theta^*, S') = J(\theta^*, S^*)$ for all $S' \in \mathcal{B}(S^*)$, i.e. the return is constant on the neighborhood.

For the third claim, Assumption 1 constrains $J(\theta_0, \cdot)$ only if θ_0 was the policy against which the bank step was solved. If it was not, no constraint relates $J(\theta_0, S^*)$ to $J(\theta_0, S')$ for $S' \in \mathcal{B}(S^*)$, so $\max \mathcal{R}(\theta_0, S^*) > 0$ is consistent with the premise. An explicit instance in which this maximum is strictly positive is given in Appendix D.3. □

Remark 1 (The one-sidedness is essential). Because $S^* \in \mathcal{B}(S^*)$, the inequality $\Phi \ge 0$ holds for every policy and carries no information about co-evolution. The content of Proposition 1 is the *sign* statement $\mathcal{R} \subseteq (-\infty, 0]$: for a co-evolved policy the response set has no positive part. Reporting only the magnitude of the worst-case drop therefore cannot distinguish a co-evolved policy from a merely brittle one, which is why the per-axis results in Section 3 report the full sign pattern rather than magnitude alone.

Remark 2 (Why removing skills is excluded). Removing every skill can *raise* a co-evolved policy's return, which would appear to contradict Eq. (5). It does not: the empty bank lies outside $\mathcal{B}(S^*)$ by Definition 1, and the exclusion is not a convenience. Assumption 1 says only that S^* beats what the proposal mechanism *offered*, and no co-evolution proposal mechanism offers to delete the bank; the empty bank was therefore never rejected in favor of S^* and nothing constrains its return. The theory's domain thus coincides with the protocol's decision to treat removal as an anchor rather than a perturbation (Section 4.2).

D.3 Construction 1 and Proof of Proposition 3

Propositions 1 and 2 bound the sign of the response but say nothing about the magnitude of Φ or its evolution during training. We exhibit the smallest instance containing both ingredients of co-evolution—a policy that reads the bank, and a bank selected under Assumption 1 to help the current policy—in which both are available in closed form.

Construction 1 (Minimal linear–Gaussian instance)**.** A binary label is predicted from two independent noisy channels: a *semantic* channel whose strength is fixed by the environment, and a *form* channel whose salience the bank step controls. The policy is a linear classifier trained as a best response, and the bank step raises form salience under Assumption 1.

The instance. A label $y \in \{-1, +1\}$ is uniform. The policy observes two scalar channels,

$$x_r \;=\; \mu_r\, y + \varepsilon_r, \quad \varepsilon_r \sim \mathcal{N}(0, \sigma_r^2), \qquad x_u \;=\; \mu_u\, y + \varepsilon_u, \quad \varepsilon_u \sim \mathcal{N}(0, \sigma_u^2), \tag{14}$$

with independent noises. The *semantic* channel x_r has strength μ_r fixed by the environment: it represents what the skill means, which no rewrite alters. The *form* channel x_u carries the label only through the particular surface encoding the bank supplies, and its salience μ_u is the quantity the bank step controls. The policy is the linear rule $\operatorname{sign}(w \cdot x)$ with $w = (w_r, w_u)$. Conditioned on y we have $w \cdot x \sim \mathcal{N}(y\, m, s^2)$ with $m = w_r\mu_r + w_u\mu_u$ and $s^2 = w_r^2\sigma_r^2 + w_u^2\sigma_u^2$, so the return is

$$J(w, \mu_u) \;=\; \Phi_{\mathcal{N}}\!\left(\frac{w_r\mu_r + w_u\mu_u}{\sqrt{w_r^2\sigma_r^2 + w_u^2\sigma_u^2}}\right), \tag{15}$$

where $\Phi_{\mathcal{N}}$ is the standard normal CDF.

The dynamics. The *policy step* maximizes Eq. (15) over w, which is the Fisher discriminant $w \propto (\mu_r/\sigma_r^2,\ \mu_u/\sigma_u^2)$. The *bank step* follows Assumption 1: since Eq. (15) is increasing in μ_u whenever $w_u > 0$, any candidate that raises form salience is accepted, so $\mu_u^{(t)}$ is non-decreasing in t. Define the two squared signal-to-noise ratios

$$a \;=\; \frac{\mu_r^2}{\sigma_r^2} \quad \text{(fixed)}, \qquad b^{(t)} \;=\; \frac{\big(\mu_u^{(t)}\big)^2}{\sigma_u^2} \quad \text{(non-decreasing; the } \textit{dose}\text{)}. \tag{16}$$

Clean and perturbed return. Substituting the Fisher weights into Eq. (15) gives numerator $\mu_r^2/\sigma_r^2 + \mu_u^2/\sigma_u^2 = a + b$ and denominator $\sqrt{\mu_r^2/\sigma_r^2 + \mu_u^2/\sigma_u^2} = \sqrt{a+b}$, hence

$$J_{\text{clean}}^{(t)} \;=\; \Phi_{\mathcal{N}}\!\left(\sqrt{a + b^{(t)}}\right). \tag{17}$$

A meaning-preserving rewrite changes the surface encoding while leaving the semantics intact, so the direction the policy learned to read no longer predicts the label: in the model, $\mu_u \to 0$ in the data while the weights, fixed at deployment, and the channel noise are unchanged. The numerator becomes a and the denominator is unchanged, giving

$$J_{\text{pert}}^{(t)} \;=\; \Phi_{\mathcal{N}}\!\left(\frac{a}{\sqrt{a + b^{(t)}}}\right), \qquad \Phi^{(t)} \;=\; \Phi_{\mathcal{N}}\!\left(\sqrt{a + b^{(t)}}\right) - \Phi_{\mathcal{N}}\!\left(\frac{a}{\sqrt{a + b^{(t)}}}\right). \tag{18}$$

Proposition 3 (Dose–response, in Construction 1)**.** *Both peak return and fragility $\Phi^{(t)}$ are strictly increasing in the dose $b^{(t)}$. A policy that does not read the form channel has $\Phi^{(t)} = 0$ for all t.*

Proof of Proposition 3. Write $\varphi_c(b) = \Phi_{\mathcal{N}}(\sqrt{a+b})$ and $\varphi_p(b) = \Phi_{\mathcal{N}}(a/\sqrt{a+b})$ for $b \geq 0$, so that $J_{\text{clean}} = \varphi_c(b)$ and $\Phi = \varphi_c(b) - \varphi_p(b)$. The map $b \mapsto \sqrt{a+b}$ is strictly increasing and $\Phi_{\mathcal{N}}$ is strictly increasing, so φ_c is strictly increasing: peak return rises with the dose. The map $b \mapsto a/\sqrt{a+b}$ is strictly decreasing for $a > 0$, since the numerator is constant and the denominator strictly increases, so φ_p is strictly decreasing. Hence $\Phi = \varphi_c - \varphi_p$ is a sum of a strictly increasing function and the negative of a strictly decreasing function, and is therefore strictly increasing in b. Because $b^{(t)}$ is non-decreasing along the trajectory and strictly increases at any accepted step, both $J_{\text{clean}}^{(t)}$ and $\Phi^{(t)}$ increase along training.

For the final claim, a policy with $w_u = 0$ never reads the form channel; then $m = w_r\mu_r$ and $s^2 = w_r^2\sigma_r^2$ both in the clean and in the perturbed condition, so $J_{\text{clean}} = J_{\text{pert}} = \Phi_{\mathcal{N}}(\sqrt{a})$ and $\Phi = 0$ for every t. □

Corollary 1 (The sign reversal of Proposition 1, realized)**.** *Let θ_0 have a misaligned form weight, $w_u\mu_u < 0$, as may happen for a policy that never co-evolved to this bank. Then the numerator of Eq.* (15) *satisfies $w_r\mu_r + w_u\mu_u < w_r\mu_r$, while the perturbed numerator is $w_r\mu_r$ and the denominator is unchanged; hence $J_{\mathrm{pert}} > J_{\mathrm{clean}}$ and $\max \mathcal{R}(\theta_0, S^*) > 0$. Destroying a channel the policy was reading incorrectly improves it, which is the admissible case of Proposition 1.*

Remark 3 (Modeling choice in the policy step). The policy step above is idealized as an exact best response, which is what makes Eq. (18) exact. Modeling it as gradient ascent is closer to practice but makes the closed form asymptotic, in which case Proposition 3 reads "increasing in the limit" rather than "strictly increasing". We keep the best-response form for its clean closed form and note the gradient-ascent reading here.

Remark 4 (What the construction does not explain). Within Construction 1 a harmful rewrite corresponds to reversing the form channel, $\mu_u \to -\mu_u$, which yields numerator $a - b$ and hence a larger degradation than the meaning-preserving rewrite for every $b > 0$. The instance therefore predicts that harmful edits hurt more than meaning-preserving ones. We do not test this ordering, and flag it as a limitation of the instance: it isolates what a two-channel account can capture and leaves finer orderings among the axes open.

D.4 Proof of Proposition 2 and the Invariance of the Point-Mass Objective

Proof of Proposition 2. Add and subtract $J(\theta, S^*)$ inside the expectation,

$$\mathbb{E}_{S\sim P}[J(\theta,S)] = \mathbb{E}_{S\sim P}\big[J(\theta,S^*) - \big(J(\theta,S^*) - J(\theta,S)\big)\big] = J(\theta,S^*) - \mathbb{E}_{S\sim P}[J(\theta,S^*) - J(\theta,S)]\,,$$

which is Eq. (6) since $J(\theta, S^*)$ does not depend on S. Under Assumptions 1 and 2, each integrand $J(\theta^*, S^*) - J(\theta^*, S)$ is non-negative by Proposition 1, so the average degradation is non-negative; it is bounded above by $\Phi(\theta^*, S^*)$ because every integrand is. Taking $P = \delta_{S^*}$ makes the integrand vanish identically, recovering $J(\theta, S^*)$. □

The companion statement referred to in Section 2.3 says that the training objective cannot see fragility at all. It is a statement about the *objective*, not about a particular network, and unlike Proposition 2 it needs a hypothesis on the policy class.

Proposition 4 (The point-mass objective is blind to fragility)**.** *Suppose the policy class is rich enough that behavior at S^* can be held fixed while behavior elsewhere on $\mathcal{B}(S^*)$ varies: for any θ_1 there exists θ_2 with $\pi_{\theta_1}(\cdot \mid x, S^*) = \pi_{\theta_2}(\cdot \mid x, S^*)$ for all x and arbitrary behavior on $\mathcal{B}(S^*) \setminus \{S^*\}$. Then the training objective $J(\cdot, S^*)$ is invariant to Φ: its maximizers span the full achievable range of fragility, so training provides no signal on it.*

Proof. Let θ_1 be any maximizer of $J(\cdot, S^*)$. By the richness hypothesis there is θ_2 agreeing with θ_1 at S^* for every task, hence with $J(\theta_2, S^*) = J(\theta_1, S^*)$, so θ_2 is also a maximizer, while its behavior on $\mathcal{B}(S^*) \setminus \{S^*\}$ is arbitrary and therefore so is $\min_{S'} J(\theta_2, S')$. Consequently $\Phi(\theta_2, S^*)$ ranges over the full achievable interval as θ_2 varies within the set of maximizers: the objective does not distinguish them, and its gradient with respect to any parameter affecting only off-S^* behavior vanishes. □

The hypothesis is what makes precise that $J(\cdot, S^*)$ cannot distinguish a fragile policy from a robust one. We do not claim both policies are reachable under a shared-weight parameterization, where behavior at S^* and on its neighborhood are coupled; this is why the main text relies on Eq. (6) alone.

Corollary 2 (Hedging in Construction 1)**.** *Take P to leave the form channel intact with probability p and to destroy it (setting $\mu_u \to 0$) otherwise, and use the scale invariance of the sign classifier to fix $w_r = 1$. With $s^2 = \sigma_r^2 + w_u^2\sigma_u^2$, the marginalized return is*

$$\bar{J}(w_u) \;=\; p\,\Phi_{\mathcal{N}}\!\left(\frac{\mu_r + w_u\mu_u}{s}\right) \;+\; (1-p)\,\Phi_{\mathcal{N}}\!\left(\frac{\mu_r}{s}\right). \tag{19}$$

Since $\frac{d}{dw_u}\frac{\mu_r+w_u\mu_u}{s} = \frac{\mu_u\sigma_r^2 - \mu_r w_u\sigma_u^2}{s^3}$ and $\frac{d}{dw_u}\frac{\mu_r}{s} = \frac{-\mu_r w_u\sigma_u^2}{s^3}$,

$$s^3\,\bar{J}'(w_u) \;=\; p\,\varphi\!\left(\frac{\mu_r + w_u\mu_u}{s}\right)\big(\mu_u\sigma_r^2 - \mu_r w_u\sigma_u^2\big) \;-\; (1-p)\,\varphi\!\left(\frac{\mu_r}{s}\right)\mu_r w_u\sigma_u^2, \tag{20}$$

where φ is the standard normal density. Let $w_u^\star = \mu_u\sigma_r^2/(\mu_r\sigma_u^2)$ be the unhedged optimum, at which the first bracket vanishes and $p = 1$ recovers the Fisher rule. For any $p < 1$ the second term is strictly negative at $w_u^\star$, so $\bar{J}'(w_u^\star) < 0$ and the maximizer satisfies $w_u^{(p)} < w_u^\star$: the policy underweights the form channel, by an amount that grows with $1 - p$. The hedged weight lowers J_{clean} and shrinks Φ, so the marginalized objective trades peak return for bounded fragility, in the direction predicted by Eq. (6).

D.5 Adversarial Co-evolution

We record the sign-flipped counterpart of Proposition 1 referred to in the Scope paragraph of Section 2.3. Replacing the cooperative gate of Eq. (2) by an adversarial one, as in solver–generator and attacker–defender training (Ke et al., 2026; Wen et al., 2026), sets $S_t \in \arg\min_{S\in\mathcal{C}_t} J(\theta_t, S)$. Then $J(\theta_t, S_t) = \min_{S\in\mathcal{C}_t} J(\theta_t, S)$, so every element of $\mathcal{R}(\theta_t, S_t)$ is non-negative and $\Phi(\theta_t, S_t) = 0$: at the retained bank, no content-level perturbation can hurt, because the retained bank already is the worst case. A policy step that subsequently raises $J(\cdot, S_t)$ therefore raises the neighborhood minimum rather than the neighborhood maximum. This is the formal sense in which the sign of the acceptance test, and not the presence of co-evolution, determines whether training pressure acts on the worst case.

E CODA Protocol Specification

E.1 Design Principles

Two principles guide the protocol. *Checkpoint-free*: CODA uses only the delivered pair plus the shared suite, because step definitions are not comparable across methods, many releases keep only the final checkpoint, and prompt-based evolution has no notion of a step at all; the fragility curve of Section 3.2 therefore serves as causal *evidence* in this paper, not a protocol metric. *Direction-first*: each metric's direction is stated before its formula, and any quantity without a defined direction serves as a gate rather than a score, as T5 does. The metric set is thus closed: diagnostics without a defined direction, such as the surface-form dependence ratio κ of Appendix H.5, are excluded from the report card.

E.2 Requirements on the Rewriting Suite (R1–R5)

Five requirements make the hard rule of Section 4.1—that rewriting is performed by the released suite, not by the party under evaluation—workable across methods with incomparable bank formats. The remaining subsections of this appendix specify each in implementable detail.

- **R1** *Selection masking:* retrieval is run once on the unperturbed bank and only the content of the selected skills is substituted (Appendix E.5).
- **R2** *Field-classification interface:* a method declares which fields are retrieval-side (frozen) and which are content (rewritable), rather than standardizing a bank format.
- **R3** *Reproducibility:* the rewriting model, its version and its sampling are pinned, and the perturbed banks—not a script—are released (Appendix E.4).
- **R4** *Structural checks, not a semantic verifier:* only declared content fields may change and the bank structure is preserved; meaning-preservation is left to the released banks for inspection, not to a similarity threshold, and is validated in Appendix H.6.
- **R5** *Strength reporting:* the suite reports the edit distance and similarity it produced, the analog of stating an attack budget.

E.3 Perturbation Operators

A skill σ is a record of typed fields, partitioned into a *retrieval-side* part $\rho(\sigma)$—the fields a method uses to decide when the skill is surfaced, such as its title, trigger conditions, keywords and usage statistics—and a *content* part $\gamma(\sigma)$, the procedural text the policy reads once the skill is in context. A bank is a finite set $S = \{\sigma_1, \ldots, \sigma_n\}$, and each axis is an operator on banks.

The content-level axes T1–T3 keep the set of skills fixed and rewrite only content fields, so their image lies in the neighborhood $\mathcal{B}(S^*)$ of Definition 1. Writing g_a for the layered-prompt rewriter of Appendix E.4 applied field-by-field to the declared content fields, with $\rho(\sigma)$ copied verbatim:

- **T1 (semantic)** replaces $\gamma(\sigma)$ by a meaning-preserving paraphrase $g_1(\gamma(\sigma))$: the wording changes, the prescribed behavior does not.
- **T2 (sufficiency)** replaces $\gamma(\sigma)$ by $g_2(\gamma(\sigma))$ that stays on topic but no longer suffices, either by deleting a required step or by injecting irrelevant material.
- **T3 (misleading)** replaces $\gamma(\sigma)$ by $g_3(\gamma(\sigma))$ that is well-formed but prescribes an incorrect strategy.

The structural axes T4–T5 change which skills are present and therefore lie outside $\mathcal{B}(S^*)$:

- **T4 (ecosystem)** applies a sequence of composition edits—insert a new skill, delete an incumbent, or replace a skill by an alternative version—leaving the text of every surviving skill unchanged.
- **T5 (autonomy)** removes the bank entirely, $T_5(S) = \varnothing$.

E.4 Layered Rewriting Prompt

Section 3.1 states that skill rewrites are produced by `gpt-4o-2024-08-06` at temperature 0.7. This appendix expands the prompt architecture that anchors the rewriter in each benchmark's semantics.

Why a layered prompt. Handing a skill to a general-purpose model with a single rewriting instruction risks two failure modes: the output may reference actions or objects that do not exist in the environment, and the edit may drift from the intended axis (e.g. a T1 "paraphrase" that silently removes a step, which is T2 territory). A layered prompt prevents both by supplying the domain context the model lacks and restricting the scope of the edit.

Architecture. The prompt is assembled from five layers (Table 3). Layers 1–4 are shared across all content-level axes and fix everything the rewrite must respect—the environment, the tokens that must survive verbatim, the task category and the skill's capability tag. Only the fifth layer carries the axis directive, so one shared skeleton plus one swapped instruction produces every content-level variant. The user turn supplies the specific skill field to be rewritten.

Field-level targeting. Each axis declares which fields it may touch and the remainder are copied verbatim. T1, for instance, rewrites a skill's `principle` and `when_to_apply` and the `description`, `why` and `how` of a recorded mistake, while titles and identifiers are left unchanged—ensuring that retrieval-side metadata is never accidentally altered.

T4 and T5. The ecosystem (T4) and autonomy (T5) axes are structural operations on the bank (adding, removing or replacing skills, or removing the bank entirely) rather than text rewrites, and do not use this prompt.

Since the same prompt need not return the same text twice, the banks we evaluated against are released alongside the suite and are the reproducible artifact.

E.5 Selection Masking

Masking separates the effect of rewritten content from the effect of changed retrieval. Let $\mathrm{Ret}_\theta(S; i)$ denote the method's retrieval step: the subset of bank S it surfaces into context for task i under policy θ. For a content-level axis $a \in \{\mathrm{T1}, \mathrm{T2}, \mathrm{T3}\}$ and each task i:

1. Run retrieval once on the unperturbed bank, $\mathrm{Sel}_i = \mathrm{Ret}_\theta(S^*; i)$, and record the ids of the selected skills.
2. Evaluate the perturbed condition without re-running retrieval: inject exactly the skills whose ids lie in Sel_i, each with its content fields replaced by the axis-a rewrite and its retrieval-side fields $\rho(\sigma)$ left untouched.

Table 3: The layered rewriting prompt for the content-level axes. Layers 1–4 form a shared skeleton; only the tier instruction (layer 5) changes between axes. The per-benchmark `domain_context` is released with the suite, so an adopter runs the prompt with no manual authoring step.

	Layer	Role	Scope
System prompt			
1	`domain_context`	Task semantics and the executable action space of the benchmark.	shared
2	`domain_terms`	Protected list: action and interface tokens preserved verbatim.	shared
3	`category_hint`	The task category the skill belongs to.	shared
4	`skill_characteristic`	The skill's capability tag, focusing the edit on the right content.	shared
5	tier instruction	The axis directive: T1 rephrase synonymously; T2 delete steps or inject irrelevant material; T3 substitute an incorrect strategy.	per axis
User turn			
	field text	The skill field to rewrite (`principle`, `when_to_apply`, ...).	per skill

Because the surfaced set is Sel_i in both the unperturbed and the perturbed condition, any change in outcome is attributable to the rewritten content rather than to a different set of skills being retrieved, and coverage is identical across methods regardless of bank size. This constancy of the selection map on $\mathcal{B}(S^*)$ is also what licenses folding selection into π_θ in Section 2, so that J can be treated as a function of the bank as a whole. Masking applies only to T1–T3. The structural axes T4 and T5 change which skills exist by design, so retrieval is re-run against the modified bank and the selected set is not held fixed; by Definition 1 these axes fall outside $\mathcal{B}(S^*)$, and the propositions of Section 2 make no claim about them.

E.6 Field-Classification Interface

An adopter declares, once per bank format, which fields the suite may rewrite (*content*) and which must remain untouched (*retrieval-side*). The content fields are the target of T1–T3; the retrieval-side fields determine which skills are selected and must be frozen so that performance changes can be attributed to content alone (Section 3.1). Two examples from this paper:

Format	Content (rewritable)	Retrieval-side (frozen)
SkillRL	`principle`, `example`	`name`, `keywords`, `embedding`
ReSkill	`strategy`	`task_type`, `score`, `embedding`

The declaration is a one-line JSON mapping; the suite validates it at run time and refuses to rewrite a field classified as retrieval-side.

E.7 Quality Control and Strength Reporting

Three checks run automatically, and it is worth being exact about what they do and do not establish.

Field targeting. Each axis declares the fields it may rewrite and the remainder are copied verbatim. T1, for instance, rewrites a skill's `principle` and `when_to_apply` and the `description`, `why` and `how` of a recorded mistake, while titles and identifiers are untouched.

Length band. The tier instruction asks for a rewrite within $\pm 30\%$ of the original length; enforcement is looser, rejecting a field only when the ratio of rewritten to original length falls outside $[0.5, 2.0]$. A rejected field is regenerated once, and the second attempt is accepted whatever its length. Separately, the pipeline retries on transport failures with exponential backoff; those retries are unrelated to quality and are not a check.

Table 4: The CODA report card: the quantities every method must publish, with the role each plays. Only the three scored rows enter the verdict of Section 4.3; the anchors are required because the scores are defined against them. Definitions are given by the equation referenced in each row.

Quantity	Symbol	Definition	Role	Direction	Axes
Peak success rate	$\mathrm{SR}(S^*)$	—	reported, not judged	—	—
No-skill success rate	$\mathrm{SR}(\varnothing)$	—	anchor	undefined	T5
Net skill benefit	Δ_{net}	Eq. (9)	gate	> 0 required	T5
Consistency rate	CR	Eq. (10)	score	higher better	T1
Resistance rate	RR_a	Eq. (11)	score	higher better	T2, T3
Worst-case SR	$\mathrm{SR}_{\min}$	Eq. (13)	ranking	higher better	T1–T3
Mean perturbed SR	$\overline{\mathrm{SR}}_{\mathrm{avg}}$	Eq. (13)	secondary	higher better	T1–T3

Structural validation of the misleading axis. The T3 banks are verified after generation to contain the same skill identifiers in the same order and the same number, to leave titles and `when_to_apply` unchanged, to have modified every `principle`, and to leave the recorded mistakes alone. This establishes that the edit landed where it was supposed to land, not that the resulting strategy is wrong.

No check tests meaning. There is no embedding-similarity threshold, no entailment test for T1 and no verification that a T3 variant genuinely prescribes an incorrect strategy; requirement R4 (Appendix E.2) gives the reasoning, and the released banks are what a reader inspects instead.

Strength reporting (R5). For each released perturbed bank, the suite logs two perturbation-strength statistics per skill: (a) the character-level edit distance between the original and rewritten text, normalized by the original length, and (b) the cosine similarity of sentence embeddings (all-MiniLM-L6-v2) between the original and rewritten versions. These distributions are released alongside the perturbed banks so that adopters can verify the perturbation intensity falls within the intended range for each axis. The suite additionally records the *prompt-share statistic*: the fraction of the inference prompt occupied by injected skill text, which contextualizes how much of the input the policy relies on from the bank.

E.8 Metric Definitions and Logging Requirements

This appendix carries out the three derivations referenced from Section 4.2, and states the per-episode logging requirement without which the paired quantities in that section cannot be computed at all.

(a) Why CR and not $|\Delta\mathrm{SR}|$. On the paired binary outcomes $(c_i(S^*), c_i(S^*_{\mathrm{T1}}))$ let $n_{10} = |\{i : c_i(S^*) = 1, c_i(S^*_{\mathrm{T1}}) = 0\}|$ and $n_{01} = |\{i : c_i(S^*) = 0, c_i(S^*_{\mathrm{T1}}) = 1\}|$ be the counts of the two directions of change. The consistency rate counts the tasks that do *not* flip, $\mathrm{CR} = 1-(n_{10}+n_{01})/N$, and

$$|\Delta\mathrm{SR}| \;=\; \frac{|n_{10} - n_{01}|}{N} \;\le\; \frac{n_{10} + n_{01}}{N} \;=\; 1 - \mathrm{CR}, \tag{21}$$

with equality only when every flip runs the same way. Score invariance is therefore strictly weaker than the behavioral invariance we are after: three tasks turning wrong and three turning right leave $\Delta\mathrm{SR} = 0$ while six behaviors have changed, and a method can achieve small $|\Delta\mathrm{SR}|$ by accident. This is why the semantic-consistency metric is stated on CR.

(b) $\mathrm{SR}_{\min}$ as a lower bound on Φ. Definition 2 sets $\Phi(\theta^*, S^*) = \max_{S' \in \mathcal{B}(S^*)} \big(\mathrm{SR}(S^*) - \mathrm{SR}(S')\big)$, the largest drop over the full content-level neighborhood. The protocol only measures the three axes T1–T3, so the maximum it can attain is $\mathrm{SR}(S^*) - \mathrm{SR}_{\min}$, taken over the three sampled banks S^*_a. Since these are elements of $\mathcal{B}(S^*)$,

$$\mathrm{SR}(S^*) - \mathrm{SR}_{\min} \;\le\; \Phi(\theta^*, S^*);$$

the gap we report is thus a lower bound on the true fragility, not the exact worst case over the full neighborhood.

Why T4 and T5 are reported but not scored. T5 (autonomy) has no defined direction: removing the bank may hurt because it held knowledge the policy could not internalize, or because it held strategies the policy should have absorbed, and the number alone cannot tell these apart; the no-skill condition therefore serves as the anchor for the gate and the resistance criterion rather than as a score. T4 (ecosystem) is not the same operator across methods—bank sizes differ by an order of magnitude, from tens or hundreds of skills to a cap of eight—so it is reported (Section 3, Appendix H) but excluded from the score, regardless of whether its effect is larger or smaller than the content-level axes in a given setting.

Consistency and its noise floor. The consistency criterion tests CR against a two-standard-deviation band around CR_0, the agreement between two unperturbed runs, which fixes the repeat-level noise floor. Rather than tabulate the per-setting standard deviations, we read the outcome from the flip rates directly: CR_0 corresponds to a 6–8% flip rate between two unperturbed evaluations—the churn attributable to decoding randomness alone—whereas the T1 rewrite flips 13–23% of task outcomes, two to three times that floor. This gap exceeds repeat-level evaluation noise in every setting, so CR sits below the band and the consistency criterion fails throughout. The per-episode logs needed to recompute CR and CR_0 are released with the suite.

Logging requirements. The identities above and McNemar's test on Eq. (12) all read paired binary outcomes at the level of individual tasks. An adopter therefore must log, for every evaluation condition (unperturbed, T1, T2, T3, and $\varnothing$) and every repeat, the per-episode indicator $c_i \in \{0, 1\}$ together with the task identifier i. Two consequences follow:

- *Common task set.* The no-skill condition, the unperturbed condition, and each axis-a condition must be evaluated on the same N task identifiers. Aggregating a success rate on a slightly different task subset breaks RR_a and the McNemar statistic, since the pairing no longer holds task-by-task.
- *Aggregated scores are not sufficient.* A report card that publishes only SR per condition, without per-episode outcomes, is not auditable: neither CR, RR_a nor the McNemar p can be recomputed from it.

The suite therefore requires per-episode logs alongside the aggregated report card, and its reference implementation writes them out by default.

E.9 Auditing Each Method on Its Own Bank

Section 4.3 states that CODA audits each method on the bank it delivered rather than on a shared bank, and gives a one-paragraph summary of why. The complete argument runs to four points.

First, the unit of evaluation is the artifact the method delivers. The bank is not an external test condition supplied by us; it is half of what the method produced. Auditing method A on A's bank measures exactly what A would deploy, whereas forcing every method onto a shared bank measures a configuration nobody ships. A shared-bank protocol would therefore answer a different question: how well does each method run on a piece of text it did not choose. That is a reasonable question, but it is not a robustness question about the delivered system.

Second, what a robustness protocol standardizes is the operator and the budget, not the perturbation. Projected gradient descent produces a different adversarial example for every model because it uses that model's gradients, and no one calls the comparison unfair: what is fixed is the attack—its algorithm and its budget—not the resulting pixel. CODA is isomorphic. The rewriting instructions, the rewriting model, and the coverage—the set of skills actually retrieved for each task—are all fixed; the resulting text differs because the banks differ. The obligation this introduces is stated as R5: the strength actually applied to each method must be reported, so that unequal severity is visible rather than deniable.

Third, the obvious way to game the protocol does not work. A method could try to survive rewriting by shipping a bank so vague that corrupting it changes nothing. But a bank vague enough to be uncorruptable is almost always vague enough to be useless, and that shows up immediately in the gate of Eq. (9): a method that clears the gate has published a bank with real content, and one that fails the gate does not qualify. The two requirements pull against each other—benefit demands that the

Table 5: T1 (semantic): synonymous rewriting. Wording changes; the prescribed behavior does not.

Setting (skill, field)	Original	Rewritten
SKILLRL-ALFWorld (`pic_002`, principle)	Whenever a needed object is visible and reachable, immediately take it before moving elsewhere—do not leave targets uncollected.	**As soon as** an **essential** object is **within sight and accessible**, **grab** it right away before **proceeding to another location**. **Avoid leaving any intended items behind.**
SKILLRL-WebShop (`foo_004`, when_to_apply)	After selecting variants and just before initiating checkout.	**Prior to starting** the checkout **process**, **once variants have been chosen**.
RESKILL-ALFWorld (`skill_d231e3`, action)	Examine the item to ensure correct identification before moving it to the target location.	**Verify** the item's **identity by examining it** before **relocating** it to the **designated spot**.
RESKILL-ScienceWorld (`skill_209e7c`, action)	After verifying results, immediately place the object in the correct box to prevent unnecessary repetition.	**Following the confirmation of** results, **promptly move** the object to the **appropriate** box to **avoid redundant actions**.

bank carry real content, robustness demands that the policy not depend on its exact wording—and a method must satisfy both with a single artifact.

Fourth, deployment fidelity points the same way. There is no shared bank in the real world: each deployment owns a library that changes on its own schedule, for reasons internal to that deployment. Perturbing each method's own bank is the faithful simulation of that situation. A shared bank would be the distortion, not the fair comparison, because it evaluates every method under a configuration none of them would have deployed.

F PERTURBATION EXAMPLES

Tables 5–7 give one representative original/rewritten pair per content-level axis for each SKILLRL/RESKILL setting, exported directly from the banks used in evaluation without manual editing, so that a reader can judge whether the rewrites are fair. To make each edit visible at a glance, rewritten or inserted text is marked in **red bold** and text that the operator ~~removed~~ is struck through. T4 (ecosystem) changes the *composition* of the bank—skills are added or removed wholesale, with no one-to-one text correspondence—and T5 (autonomy) removes the bank entirely, so neither has a before/after text pair to show.

G EXPERIMENTAL CONFIGURATION

This appendix expands the setup summarized in Section 3.3.

Method and benchmark diversity. The four methods spread across Table 1 rather than clustering in one corner of it: they differ in the update schedule, the bank growth policy, the admission rule and the injection mechanism, sharing only the current-policy selection bias abstracted by Assumption 1. The benchmarks are equally spread, spanning embodied interaction (ALFWorld), shopping (WebShop), scientific procedures (ScienceWorld), open-web navigation (Mind2Web) and competition mathematics (AMC23); the last also tests whether the pattern extends from action procedures to non-interactive reasoning strategies, so the trend is not an artifact of sequential control.

Backbones. Each method is run on the base model used in its experimental configuration: Qwen2.5-7B-Instruct for SKILLRL and EVOLVING-RL, and Qwen3-4B-Instruct-2507 for

Table 6: T2 (sufficiency): a skill stays on topic but no longer suffices, either because a step is deleted (struck through in the original) or because irrelevant material is injected (marked in the rewrite).

Setting (skill, field, operator)	Original	Rewritten
SKILLRL-ALFWorld (`loo_005`, principle, delete)	Avoid toggling the same lamp or re-examining the same empty spot; ~~if no progress after one full check, relocate to a new plausible lamp location.~~	Avoid toggling the same lamp or re-examining the same empty spot.
SKILLRL-WebShop (`dyn_015`, principle, inject)	Relax non-essential constraints in the query based on context when early attempts yield zero or irrelevant results.	Relax non-essential constraints in the query based on context when early attempts yield zero or irrelevant results. **Consider the time of day you are searching, as it may affect product availability, and keep in mind the color of the product, even if it is not a requirement.**
RESKILL-ALFWorld (`skill_059fd6`, action, delete)	~~Always heat the item using a microwave or stoveburner before~~ placing it anywhere.	**Place** the item anywhere.
RESKILL-ScienceWorld (`skill_668201`, action, inject)	Perform a conductivity test after connecting components ~~to confirm the circuit is complete.~~	Perform a conductivity test after connecting components, **and consider noting the color of the wires used.**

Table 7: T3 (misleading): the strategy a skill prescribes is replaced by an incorrect one. Shown at the stronger of the two strengths, which is the setting reported as T3 in the main text for WebShop and ScienceWorld.

Setting (skill, field)	Original	Rewritten
SKILLRL-ALFWorld (`gen_008`, principle)	Record which locations and containers have been fully inspected; do not revisit them unless new evidence suggests their contents changed.	**Always opt for the first plausible location or container you encounter**, assuming it holds the target object, and **never backtrack or verify even if the task fails**.
SKILLRL-WebShop (`gen_011`, principle)	After exhaustive search and reasonable refinements yield no matching product, stop and report failure instead of endless retries.	**Quickly choose the first result that seems remotely relevant** and proceed to buy, **even if it does not fully match** the user's specifications or constraints.
RESKILL-ALFWorld (`skill_2a1e35`, action)	Ensure the item is cooled using the fridge before moving or placing it anywhere. Do not skip this step.	Move the item directly to any available container **without cooling it first**. **Skip any cooling step.**
RESKILL-ScienceWorld (`skill_e22fd6`, action)	Complete the circuit, verify conductivity, then decide placement to ensure accuracy.	**Assume conductivity after a single connection** and proceed with placement, **prioritizing speed over accuracy**.

RESKILL and ARISE. Because the backbones differ, an absolute success rate characterizes a method together with its backbone rather than the method in isolation.

Evaluation sets. Evaluation uses the standard split of each benchmark—140 tasks on ALFWorld, 500 on WebShop, 155 validation points on ScienceWorld, and the standard evaluation splits of

Mind2Web and DeepScaleR-AMC23—held fixed across every perturbation axis and every anchor, so the conditions being compared are paired at the level of individual tasks. The released configuration records the exact identifiers and sample counts for all six settings. Checkpoints are saved at a regular interval along each training trajectory; the exact anchor steps are shown in the corresponding fragility curves and verified full-results tables.

Repeats and error bars. Every anchor is evaluated repeatedly under six conditions—unperturbed and T1–T5—and reported with mean ± 1 s.d. The SKILLRL and RESKILL settings use three repeats, and RESKILL-ScienceWorld uses five; repeat counts for EVOLVING-RL and ARISE are recorded in the released configuration. Repeats share the policy checkpoint and the same fixed set of perturbed banks and differ only in the rollout sampling seed. The delivered-pair report cards use the paired runs required to compute CR, CR_0 and RR. Each setting comes from a single co-evolution run, so the bands quantify evaluation noise, not variability across training seeds.

Misleading-axis strength. For the four SKILLRL/RESKILL settings, the misleading axis (T3) is applied at one strength per setting: the milder variant on the two ALFWorld settings and the stronger variant on WebShop and ScienceWorld. Both strengths were run in each of these four settings, and the one not reported in the main text is included in the full-results tables below.

H FULL RESULTS OF THE CO-EVOLUTION FRAGILITY CURVES

H.1 PER-METHOD, PER-ENVIRONMENT, PER-AXIS TABLES

Tables 8–13 give the raw success rates behind all six settings in Section 3: one row per anchor and one column per condition. Every table reports the mean ± 1 s.d. over repeated evaluations. The perturbed banks are fixed across anchors, so reading down a column shows what a single fixed corruption costs a progressively more co-evolved policy. Small endpoint differences between the report card and these grids reflect repeat-level evaluation variation. All headline audit quantities use the report-card values.

Across the eighteen content-level axis–setting pairs, twelve OLS slopes of Drop are significantly positive—four on each of T1, T2 and T3. At least one axis is significant in five of six settings; SKILLRL-WebShop is the sole exception. For T3 specifically, the slope is significant on SKILLRL-ALFWorld ($p < 10^{-6}$), RESKILL-ScienceWorld ($p = 0.021$), RESKILL-ALFWorld ($p = 0.001$) and EVOLVING-RL-Mind2Web ($p = 0.041$), but not on SKILLRL-WebShop ($p = 0.351$) or ARISE-AMC23 ($p = 0.175$).

Dips inside the analyzed segment. Section 3.2 retains every anchor up to the peak, dips included; this is where they are named. Rating each dip by $|\Delta\mathrm{SR}|$ against the pooled standard deviation of the two anchors it joins, SKILLRL-ALFWorld contributes three ($\texttt{s0}\to\texttt{s5}$, -3.1 points, 2.1σ; $\texttt{s20}\to\texttt{s25}$, -8.1, 3.0σ; $\texttt{s40}\to\texttt{s45}$, -9.7, 4.4σ), RESKILL-ScienceWorld one ($\texttt{s180}\to\texttt{s240}$, -12.6, 8.1σ), RESKILL-ALFWorld two ($\texttt{s180}\to\texttt{s210}$, -21.0, 13.7σ; and $\texttt{s210}\to\texttt{s240}$, -13.1, 17.2σ, after which the run recovers and sets its highest score at the final anchor), SKILLRL-WebShop one that is indistinguishable from evaluation noise ($\texttt{s40}\to\texttt{s50}$, -1.1, 0.6σ), and EVOLVING-RL-Mind2Web two equally small reversals ($\texttt{s9}\to\texttt{s14}$, -0.1, 0.1σ; $\texttt{s19}\to\texttt{s24}$, -0.3, 0.2σ). ARISE-AMC23 is monotone at the saved anchors. We retain every dip: the dose is the amount of co-evolution a run has undergone, and a run that briefly loses ground has still undergone it. No run in the six settings peaks and then stays down, so the clause of Section 3.2 that discards post-peak checkpoints never has to be applied. Throughout, a standard deviation is the sample standard deviation over the individual evaluation repeats.

The same grids as heatmaps. Figures 5–10 display the derived quantities of Eqs. (7)–(8) over the full anchor range. Each figure has a Drop panel and an RP panel; the color scales are clipped so that the degradation regime remains legible, because the earliest anchors reach values an order of magnitude larger than the rest and would otherwise flatten the whole map into a single shade. Every cell is derived from the per-anchor means in the corresponding full-grid table, and AVG is the mean over the three scored content-level axes (T1–T3). The transition from blue on the left to white

Table 8: Full success-rate grid for SKILLRL on ALFWorld: every anchor on the rising segment, under the unperturbed bank and all five perturbation axes. Cells report mean ± 1 s.d. over 3 evaluation repeats (%). The perturbed variants are fixed across anchors; only the policy checkpoint varies.

Anchor	Unperturbed	T1	T2	T3	T4	T5
s0	$5.2_{\pm 1.5}$	$5.7_{\pm 2.5}$	$5.0_{\pm 1.2}$	$12.1_{\pm 1.7}$	$3.6_{\pm 1.7}$	$21.7_{\pm 1.2}$
s5	$2.1_{\pm 0.0}$	$2.1_{\pm 0.6}$	$2.6_{\pm 0.3}$	$7.1_{\pm 1.5}$	$4.3_{\pm 1.2}$	$18.8_{\pm 1.9}$
s10	$7.9_{\pm 2.1}$	$11.0_{\pm 0.7}$	$7.9_{\pm 2.1}$	$14.8_{\pm 0.3}$	$10.5_{\pm 1.8}$	$29.0_{\pm 1.2}$
s15	$11.0_{\pm 2.4}$	$17.6_{\pm 1.2}$	$10.5_{\pm 0.3}$	$17.1_{\pm 1.5}$	$18.3_{\pm 2.9}$	$26.4_{\pm 1.5}$
s20	$21.4_{\pm 0.6}$	$27.9_{\pm 1.5}$	$21.4_{\pm 0.0}$	$24.5_{\pm 1.2}$	$29.5_{\pm 3.0}$	$27.9_{\pm 2.1}$
s25	$13.3_{\pm 2.6}$	$22.6_{\pm 3.3}$	$11.9_{\pm 3.2}$	$20.0_{\pm 0.6}$	$21.7_{\pm 2.6}$	$23.3_{\pm 2.2}$
s30	$29.3_{\pm 2.0}$	$29.0_{\pm 1.5}$	$28.6_{\pm 4.0}$	$33.1_{\pm 2.4}$	$33.8_{\pm 3.0}$	$42.4_{\pm 0.9}$
s35	$38.1_{\pm 3.9}$	$34.5_{\pm 1.8}$	$33.8_{\pm 2.2}$	$41.7_{\pm 3.4}$	$38.6_{\pm 2.1}$	$41.9_{\pm 1.5}$
s40	$55.2_{\pm 0.9}$	$50.7_{\pm 1.5}$	$48.8_{\pm 3.7}$	$52.6_{\pm 1.2}$	$55.7_{\pm 1.5}$	$46.0_{\pm 2.8}$
s45	$45.5_{\pm 2.0}$	$42.6_{\pm 2.6}$	$46.0_{\pm 0.7}$	$46.7_{\pm 0.9}$	$41.2_{\pm 3.2}$	$39.3_{\pm 2.0}$
s50	$59.8_{\pm 2.8}$	$52.9_{\pm 3.5}$	$59.0_{\pm 4.4}$	$61.0_{\pm 2.4}$	$57.9_{\pm 4.1}$	$44.8_{\pm 2.9}$
s55	$62.6_{\pm 1.2}$	$56.4_{\pm 4.1}$	$63.8_{\pm 1.5}$	$63.8_{\pm 1.8}$	$65.0_{\pm 2.1}$	$51.0_{\pm 1.5}$
s60	$71.2_{\pm 2.0}$	$65.4_{\pm 2.4}$	$67.0_{\pm 0.7}$	$71.1_{\pm 2.9}$	$67.3_{\pm 2.1}$	$61.4_{\pm 1.0}$
s70	$83.3_{\pm 2.4}$	$75.1_{\pm 1.2}$	$76.0_{\pm 1.2}$	$76.3_{\pm 0.6}$	$72.5_{\pm 1.8}$	$63.6_{\pm 2.1}$
s80	$88.3_{\pm 1.2}$	$80.0_{\pm 2.2}$	$79.1_{\pm 1.3}$	$78.9_{\pm 2.0}$	$74.7_{\pm 2.3}$	$71.9_{\pm 2.0}$
s90	$88.8_{\pm 2.1}$	$77.6_{\pm 0.9}$	$78.6_{\pm 1.0}$	$77.1_{\pm 2.9}$	$77.6_{\pm 2.2}$	$73.3_{\pm 2.8}$

Table 9: Full success-rate grid for SKILLRL on WebShop: every anchor on the rising segment, under the unperturbed bank and all five perturbation axes. Cells report mean ± 1 s.d. over 3 evaluation repeats (%). The perturbed variants are fixed across anchors; only the policy checkpoint varies.

Anchor	Unperturbed	T1	T2	T3	T4	T5
s0	$19.2_{\pm 3.4}$	$25.3_{\pm 2.2}$	$17.1_{\pm 0.8}$	$25.9_{\pm 1.3}$	$20.3_{\pm 1.5}$	$20.3_{\pm 0.7}$
s10	$33.9_{\pm 0.9}$	$38.3_{\pm 1.9}$	$34.9_{\pm 2.0}$	$31.2_{\pm 1.4}$	$33.9_{\pm 2.8}$	$30.4_{\pm 1.9}$
s20	$38.3_{\pm 1.2}$	$40.3_{\pm 1.6}$	$39.5_{\pm 1.7}$	$37.7_{\pm 1.7}$	$39.2_{\pm 0.6}$	$33.5_{\pm 2.0}$
s30	$46.4_{\pm 2.0}$	$44.5_{\pm 2.7}$	$45.9_{\pm 1.3}$	$41.7_{\pm 0.9}$	$44.1_{\pm 3.2}$	$39.3_{\pm 0.6}$
s40	$48.4_{\pm 0.8}$	$51.5_{\pm 1.8}$	$49.3_{\pm 2.5}$	$47.9_{\pm 3.6}$	$48.7_{\pm 2.3}$	$41.6_{\pm 0.7}$
s50	$47.3_{\pm 1.8}$	$46.7_{\pm 1.2}$	$46.7_{\pm 0.6}$	$45.0_{\pm 2.0}$	$47.7_{\pm 2.3}$	$37.3_{\pm 0.5}$
s70	$52.4_{\pm 1.1}$	$51.5_{\pm 1.2}$	$48.9_{\pm 1.5}$	$51.6_{\pm 1.6}$	$52.4_{\pm 0.6}$	$45.9_{\pm 0.7}$

and red on the right is the dependence forming: perturbations that help a policy which has not yet co-evolved to the bank stop helping, and then start hurting.

H.2 CO-EVOLUTION FRAGILITY CURVES, ALL SIX SETTINGS

Figure 4 in the main text reports RESKILL on ALFWorld. The other five settings follow in the same layout: SKILLRL on WebShop (Figure 11) and ALFWorld (Figure 12), RESKILL on ScienceWorld (Figure 13), EVOLVING-RL on Mind2Web (Figure 14) and ARISE on AMC23 (Figure 15).

H.3 DROP AND RETENTION ARE ONE MEASUREMENT ON TWO SCALES

The two quantities plotted in every dose figure are algebraically linked. Substituting the definition of Drop_a into that of RP_a,

$$\mathrm{RP}_a(t) \;=\; \frac{\mathrm{SR}(S_a^*)}{\mathrm{SR}(S^*)} \;=\; \frac{\mathrm{SR}(S^*) - \mathrm{Drop}_a(t)}{\mathrm{SR}(S^*)} \;=\; 1 - \frac{\mathrm{Drop}_a(t)}{\mathrm{SR}(S^*)}, \tag{22}$$

so retention is an affine transform of the drop: a flip about the horizontal axis, rescaled by $1/\mathrm{SR}(S^*)$. Whenever the unperturbed rate is close to constant across anchors, the retention row is therefore an almost exact mirror image of the drop row, and the sole source of divergence between the two is variation in $\mathrm{SR}(S^*)$ itself.

Table 10: Full success-rate grid for RESKILL on ScienceWorld: every anchor on the rising segment, under the unperturbed bank and all five perturbation axes. Cells report mean ± 1 s.d. over 5 evaluation repeats (%). The perturbed variants are fixed across anchors; only the policy checkpoint varies.

Anchor	Unperturbed	T1	T2	T3	T4	T5
s60	$28.5_{\pm 1.8}$	$13.3_{\pm 2.5}$	$32.6_{\pm 1.4}$	$45.4_{\pm 1.8}$	$21.2_{\pm 1.7}$	$45.8_{\pm 1.9}$
s120	$36.3_{\pm 3.0}$	$37.9_{\pm 2.1}$	$22.6_{\pm 3.0}$	$41.8_{\pm 1.6}$	$48.5_{\pm 3.5}$	$36.8_{\pm 3.3}$
s180	$49.0_{\pm 0.6}$	$47.0_{\pm 2.1}$	$47.7_{\pm 1.1}$	$49.4_{\pm 1.0}$	$49.3_{\pm 1.2}$	$32.6_{\pm 3.2}$
s240	$36.4_{\pm 1.4}$	$2.7_{\pm 1.7}$	$17.3_{\pm 2.1}$	$32.3_{\pm 2.6}$	$46.1_{\pm 1.3}$	$32.1_{\pm 1.5}$
s300	$50.3_{\pm 0.0}$	$40.1_{\pm 2.1}$	$44.5_{\pm 1.2}$	$46.2_{\pm 0.3}$	$48.0_{\pm 0.9}$	$33.9_{\pm 1.7}$

Table 11: Full success-rate grid for RESKILL on ALFWorld: every anchor on the rising segment, under the unperturbed bank and all five perturbation axes. Cells report mean ± 1 s.d. over 3 evaluation repeats (%). The perturbed variants are fixed across anchors; only the policy checkpoint varies.

Anchor	Unperturbed	T1	T2	T3	T4	T5
s30	$20.0_{\pm 2.3}$	$21.2_{\pm 1.2}$	$25.2_{\pm 1.5}$	$26.2_{\pm 1.8}$	$23.3_{\pm 2.6}$	$33.6_{\pm 3.2}$
s60	$33.8_{\pm 1.8}$	$36.4_{\pm 0.6}$	$34.8_{\pm 0.9}$	$34.8_{\pm 2.2}$	$35.7_{\pm 0.6}$	$37.1_{\pm 2.1}$
s90	$37.6_{\pm 0.7}$	$35.7_{\pm 0.6}$	$35.0_{\pm 1.2}$	$39.3_{\pm 2.0}$	$38.8_{\pm 4.4}$	$41.2_{\pm 3.5}$
s120	$41.4_{\pm 1.0}$	$40.5_{\pm 1.2}$	$40.7_{\pm 0.6}$	$46.0_{\pm 1.2}$	$46.2_{\pm 2.4}$	$46.7_{\pm 1.2}$
s150	$54.8_{\pm 3.8}$	$52.9_{\pm 1.0}$	$52.6_{\pm 0.7}$	$55.0_{\pm 1.5}$	$54.0_{\pm 0.7}$	$56.4_{\pm 1.0}$
s180	$61.0_{\pm 1.5}$	$60.7_{\pm 1.0}$	$59.8_{\pm 0.9}$	$61.9_{\pm 1.2}$	$60.0_{\pm 0.6}$	$58.8_{\pm 0.3}$
s210	$40.0_{\pm 0.3}$	$36.6_{\pm 0.3}$	$38.4_{\pm 0.7}$	$39.3_{\pm 0.3}$	$39.8_{\pm 1.5}$	$38.6_{\pm 0.9}$
s240	$26.9_{\pm 0.7}$	$28.1_{\pm 0.3}$	$27.4_{\pm 0.3}$	$29.3_{\pm 1.2}$	$28.6_{\pm 0.6}$	$28.6_{\pm 1.0}$
s270	$41.4_{\pm 0.7}$	$41.1_{\pm 1.5}$	$40.4_{\pm 1.0}$	$42.6_{\pm 0.6}$	$38.8_{\pm 0.7}$	$41.9_{\pm 1.5}$
s300	$56.1_{\pm 0.6}$	$50.7_{\pm 0.0}$	$51.2_{\pm 0.3}$	$50.5_{\pm 0.3}$	$49.8_{\pm 0.3}$	$48.5_{\pm 0.3}$
s330	$71.5_{\pm 0.3}$	$62.2_{\pm 1.7}$	$61.0_{\pm 0.9}$	$63.0_{\pm 0.9}$	$61.3_{\pm 1.5}$	$64.4_{\pm 1.3}$
s360	$76.9_{\pm 0.6}$	$64.6_{\pm 0.9}$	$65.4_{\pm 2.3}$	$63.3_{\pm 2.7}$	$59.4_{\pm 0.9}$	$62.0_{\pm 1.2}$

A sign reversal along training. On SKILLRL-ALFWorld, $\mathrm{RP}_{\mathrm{T3}} = 2.33$ at `s0` and 3.38 at `s5`: a misleading bank more than doubles the score of a policy that has not yet co-evolved with it. By `s90`, $\mathrm{RP}_{\mathrm{T3}}$ falls to 0.86, while T5 moves from 4.17 to 0.82. The same bank that initially obstructs the policy thus becomes a source of competence and, eventually, dependence.

Across the six repeatedly evaluated settings, Figure 13 spans a relatively narrow competence range (28.5% to 50.3%), so its drop and retention rows mirror each other almost perfectly. Figure 12 spans 5.2% to 88.8%; there the small early denominators stretch retention onto a different scale altogether—T5 exceeds $\mathrm{RP} = 7$ at the base anchor—and the two rows take visibly different shapes.

We report both regardless, for opposite reasons: the absolute drop is what a deployment actually loses, while retention is the only one of the two that remains comparable across checkpoints—or across settings—whose ability differs by an order of magnitude. At a fixed anchor neither carries information the other lacks; what retention adds is comparability along the dose axis.

H.4 DEPENDENCE MEDIATES FRAGILITY

The dose curves plot fragility against the co-evolution step, but the step is only a proxy: what should actually govern how much a fixed corruption costs is how much the policy has come to *rely* on the bank. Figure 16 tests this directly. For every checkpoint we place one point at (x, y) with x the dependence—the drop in SR when the bank is removed entirely (the no-skill / autonomy contrast)—and y the fragility—the mean Drop over the five corruption tiers. Reading fragility against measured dependence removes the proxy, which matters because on SKILLRL-WebShop the proxy fails outright: dependence there is not monotone in the step (it reverses across an `s80`–`s110` valley), so dependence and step are essentially uncorrelated, yet the dependence–fragility relation still holds.

Table 12: Full action-success grid for EVOLVING-RL on Mind2Web: every anchor on the rising segment, under the unperturbed 200-skill bank and all five perturbation axes. Cells report mean $\pm$ 1 s.d. over evaluation repeats. The perturbed variants are fixed across anchors; only the policy checkpoint varies. T5 is the no-skill arm.

Anchor	Unperturbed	T1	T2	T3	T4	T5
s9	$16.4_{\pm 0.9}$	$17.3_{\pm 1.0}$	$21.1_{\pm 1.4}$	$15.4_{\pm 0.9}$	$18.1_{\pm 1.0}$	$14.0_{\pm 1.1}$
s14	$16.3_{\pm 0.6}$	$16.4_{\pm 1.0}$	$20.5_{\pm 1.1}$	$17.7_{\pm 1.3}$	$17.8_{\pm 0.5}$	$13.0_{\pm 0.7}$
s19	$17.1_{\pm 0.5}$	$16.5_{\pm 1.3}$	$17.1_{\pm 1.2}$	$20.1_{\pm 0.5}$	$16.5_{\pm 1.5}$	$15.8_{\pm 1.5}$
s24	$16.8_{\pm 1.1}$	$18.4_{\pm 1.1}$	$16.3_{\pm 0.6}$	$18.5_{\pm 0.4}$	$16.1_{\pm 1.0}$	$15.0_{\pm 0.5}$
s29	$18.1_{\pm 0.6}$	$17.5_{\pm 0.7}$	$18.9_{\pm 0.4}$	$19.0_{\pm 0.9}$	$17.1_{\pm 0.9}$	$13.7_{\pm 1.3}$
s34	$19.2_{\pm 1.0}$	$18.6_{\pm 1.1}$	$19.3_{\pm 1.0}$	$17.9_{\pm 1.1}$	$16.1_{\pm 0.9}$	$11.2_{\pm 0.7}$
s39	$19.8_{\pm 1.5}$	$15.7_{\pm 1.5}$	$19.4_{\pm 1.3}$	$20.5_{\pm 1.2}$	$14.7_{\pm 0.8}$	$11.0_{\pm 0.7}$
s44	$21.9_{\pm 0.7}$	$19.8_{\pm 0.5}$	$17.4_{\pm 1.2}$	$22.4_{\pm 0.8}$	$19.9_{\pm 1.3}$	$14.1_{\pm 0.8}$
s49	$23.6_{\pm 1.5}$	$21.6_{\pm 1.3}$	$19.0_{\pm 0.4}$	$20.1_{\pm 0.6}$	$19.9_{\pm 1.4}$	$13.6_{\pm 0.9}$
s54	$25.2_{\pm 1.5}$	$23.2_{\pm 0.8}$	$20.3_{\pm 0.5}$	$20.4_{\pm 1.1}$	$20.5_{\pm 1.3}$	$15.8_{\pm 0.7}$

Table 13: Full success-rate grid for ARISE on AMC23: every anchor on the analyzed segment, under the unperturbed 110-skill bank (step 2000) and all five perturbation axes. Cells report mean $\pm$ 1 s.d. over evaluation repeats. The perturbed variants are fixed across anchors; only the policy checkpoint varies. T5 is the no-skill arm.

Anchor	Unperturbed	T1	T2	T3	T4	T5
s100	$15.5_{\pm 0.8}$	$16.0_{\pm 0.6}$	$16.0_{\pm 1.1}$	$16.8_{\pm 0.5}$	$17.2_{\pm 1.0}$	$17.2_{\pm 0.8}$
s200	$17.5_{\pm 0.5}$	$18.6_{\pm 1.0}$	$18.9_{\pm 0.4}$	$16.9_{\pm 0.9}$	$18.3_{\pm 0.5}$	$19.4_{\pm 0.5}$
s300	$20.0_{\pm 0.9}$	$20.0_{\pm 1.3}$	$20.0_{\pm 0.5}$	$20.0_{\pm 0.7}$	$20.6_{\pm 1.1}$	$20.9_{\pm 1.4}$
s400	$22.5_{\pm 1.0}$	$23.6_{\pm 0.8}$	$22.9_{\pm 1.5}$	$23.9_{\pm 0.5}$	$25.0_{\pm 1.3}$	$23.9_{\pm 0.7}$
s500	$26.7_{\pm 0.6}$	$23.3_{\pm 0.5}$	$25.8_{\pm 0.7}$	$27.1_{\pm 1.3}$	$25.8_{\pm 0.6}$	$27.1_{\pm 1.0}$
s600	$31.7_{\pm 1.1}$	$30.0_{\pm 0.8}$	$31.5_{\pm 1.0}$	$32.5_{\pm 0.5}$	$34.0_{\pm 0.5}$	$32.5_{\pm 0.6}$
s700	$55.0_{\pm 1.2}$	$53.3_{\pm 0.9}$	$50.8_{\pm 0.8}$	$54.2_{\pm 1.0}$	$52.5_{\pm 0.9}$	$52.5_{\pm 0.7}$
s800	$71.9_{\pm 1.3}$	$68.8_{\pm 1.2}$	$66.7_{\pm 0.7}$	$72.9_{\pm 1.0}$	$61.5_{\pm 1.0}$	$67.7_{\pm 1.4}$
s900	$78.4_{\pm 1.2}$	$75.0_{\pm 0.7}$	$72.7_{\pm 1.5}$	$79.5_{\pm 0.5}$	$67.0_{\pm 0.9}$	$73.9_{\pm 1.2}$
s1000	$87.4_{\pm 0.6}$	$82.5_{\pm 0.9}$	$81.5_{\pm 0.4}$	$80.9_{\pm 1.1}$	$76.5_{\pm 1.2}$	$75.3_{\pm 1.0}$

The two quantities lie on a single increasing line. Pooling the 32 checkpoints across both settings gives Spearman $\rho = +0.82$ (95% CI $[+0.68, +0.89]$, $p = 7.4 \times 10^{-9}$) and an ordinary-least-squares slope of $+0.20$ pp of fragility per pp of dependence (95% bootstrap CI $[+0.15, +0.28]$, excluding zero); each setting is individually significant (SKILLRL-ALFWorld $\rho = +0.78$, $p = 1.5 \times 10^{-4}$, $n = 18$; SKILLRL-WebShop $\rho = +0.79$, $p = 7.9 \times 10^{-4}$, $n = 14$). The relation is not carried by a few extreme points: the shaded band is a 4000-sample bootstrap over anchors and stays narrow throughout. Crucially the line passes through the third quadrant. At the earliest checkpoints dependence is negative—removing the bank *raises* SR, because the policy has not yet co-evolved to it—and there fragility is negative too: corrupting a bank the policy does not rely on *helps*. The same axis that predicts vulnerability at high dependence predicts a benefit at negative dependence, which is exactly the sign structure of Proposition 1. Dependence, not elapsed training, is what fragility tracks; the dose effect of Section 3.2 is the composition of two links, training builds dependence and dependence creates fragility.

H.5 SURFACE-FORM DEPENDENCE AND PROMPT SHARE

How much of the drop is about wording? Comparing the two ends of the axis ordering gives a cheap diagnostic of surface-form dependence,

$$\kappa \;=\; \frac{\mathrm{SR}(S^*) - \mathrm{SR}(S^*_{\mathrm{T1}})}{\mathrm{SR}(S^*) - \mathrm{SR}(S^*_{\mathrm{T3}})}, \tag{23}$$

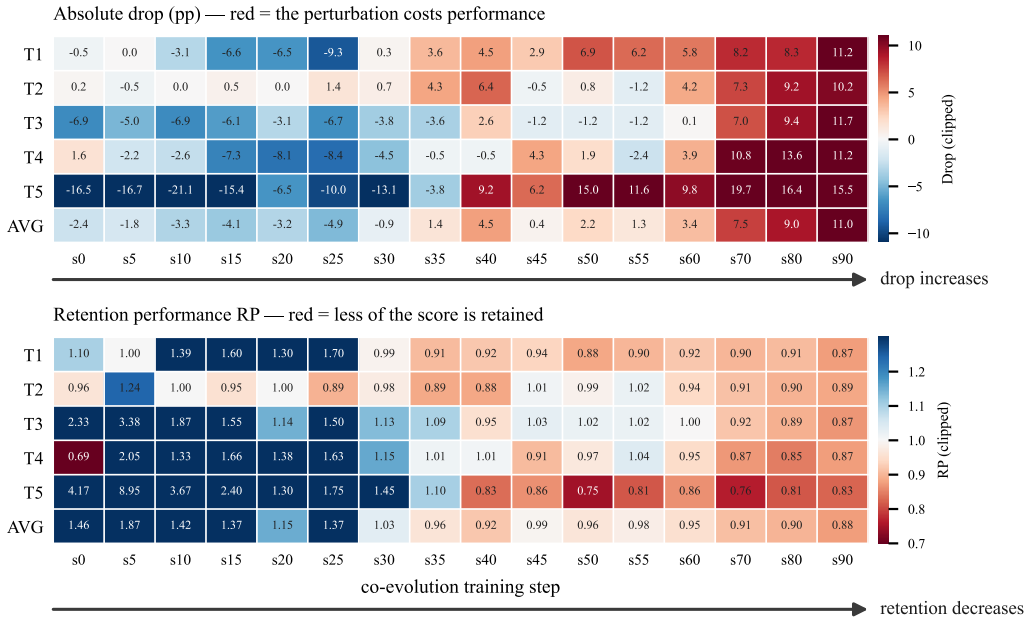


Figure 5: Perturbation response across all checkpoints, SKILLRL on ALFWorld. Top: Drop in percentage points. Bottom: retention performance RP. Color scales clipped; cell values exact.

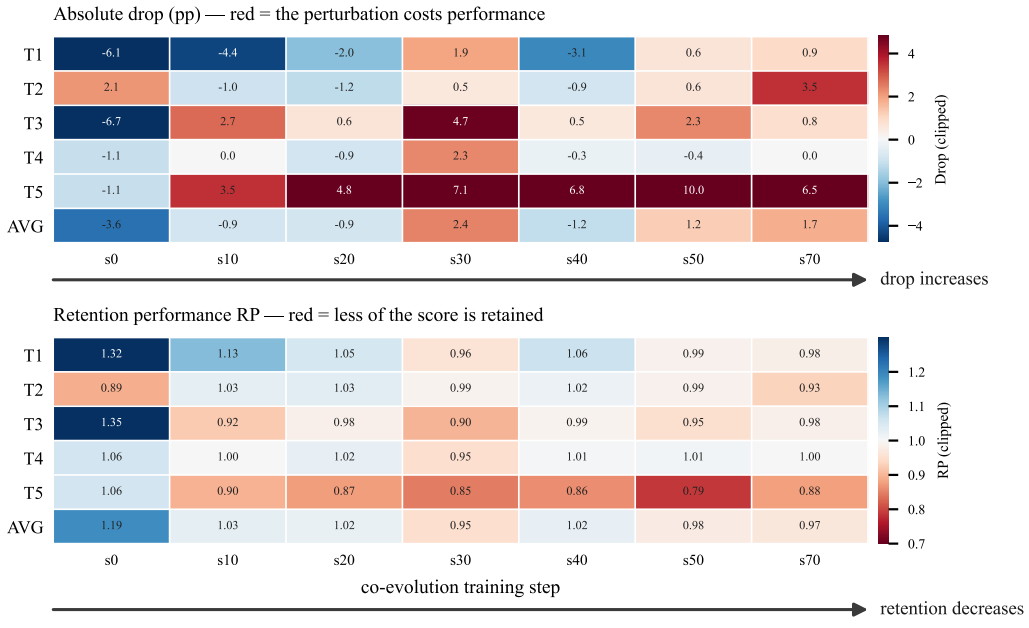


Figure 6: Perturbation response across all checkpoints, SKILLRL on WebShop. Panels and conventions as in Figure 5.

the damage done by rewriting the words relative to the damage done by rewriting the strategy. A policy that reads a skill for its meaning should have κ near zero, since a synonymous rewrite costs it nothing while a wrong strategy costs it a great deal; a policy using the bank as a look-up table has κ near one or above. Measured values are $\kappa = 0.78$ on SKILLRL-ALFWorld and 2.14 on

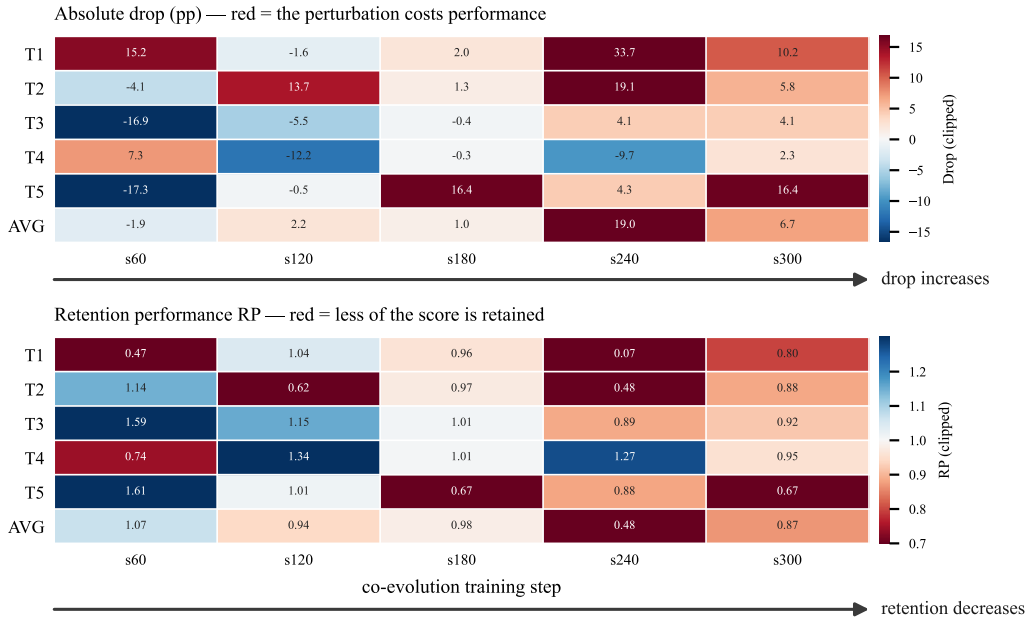


Figure 7: Perturbation response across all checkpoints, RESKILL on ScienceWorld. Panels and conventions as in Figure 5.

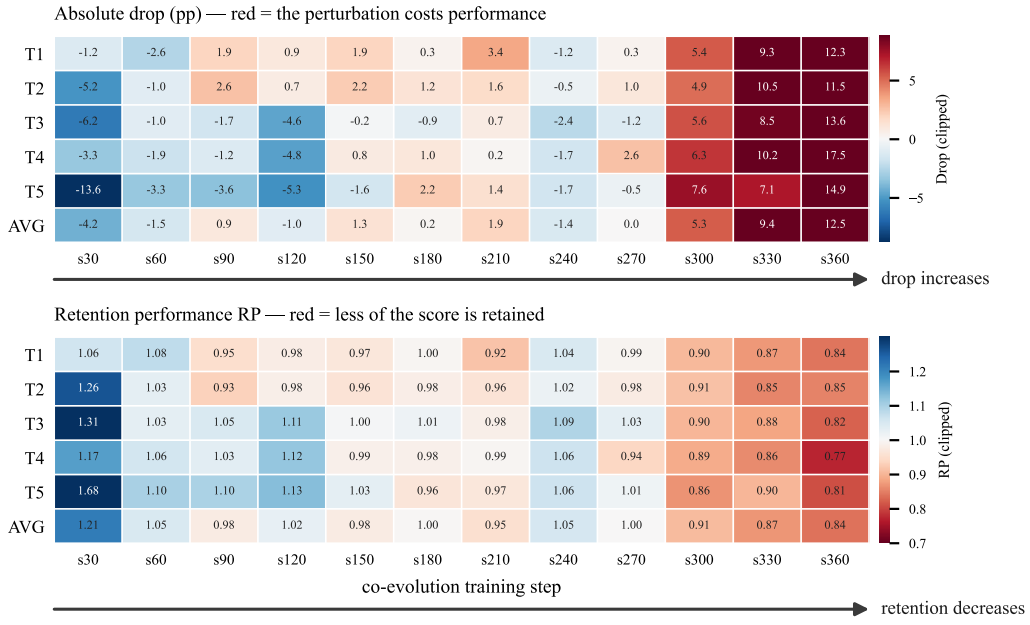


Figure 8: Perturbation response across all checkpoints, RESKILL on ALFWorld. Panels and conventions as in Figure 5.

RESKILL-ScienceWorld; the latter exceeds one, meaning a meaning-preserving change hurts more than a wrong strategy does. On the other two SKILLRL/RESKILL settings, the T3 axis does not lower the success rate below the peak, so κ is undefined (negative denominator). We report κ as

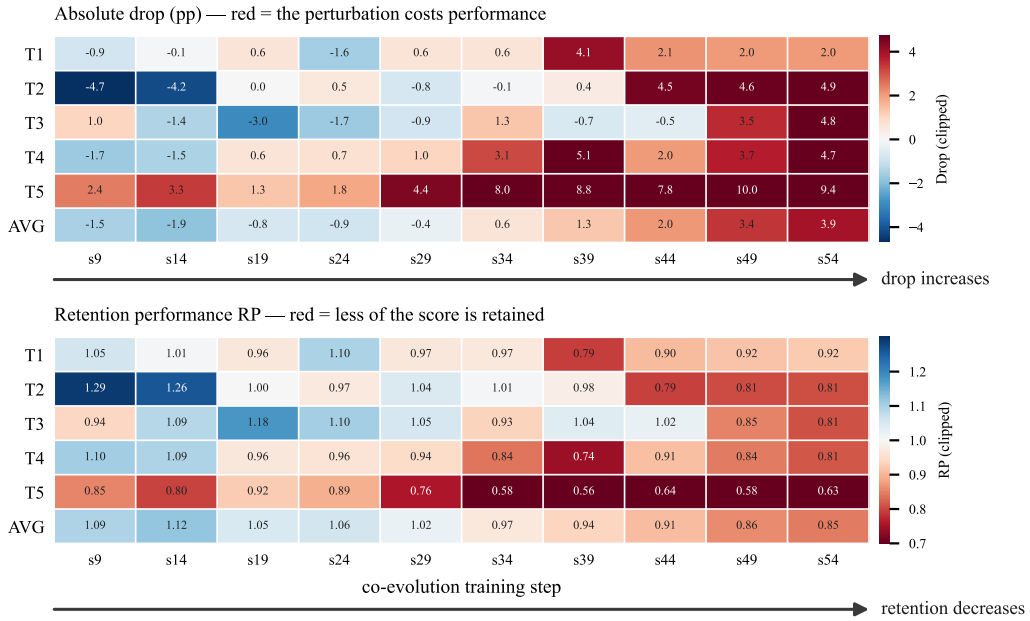


Figure 9: Perturbation response across all checkpoints, EVOLVING-RL on Mind2Web. Panels and conventions as in Figure 5; cell values are derived from the per-anchor means in Table 12.

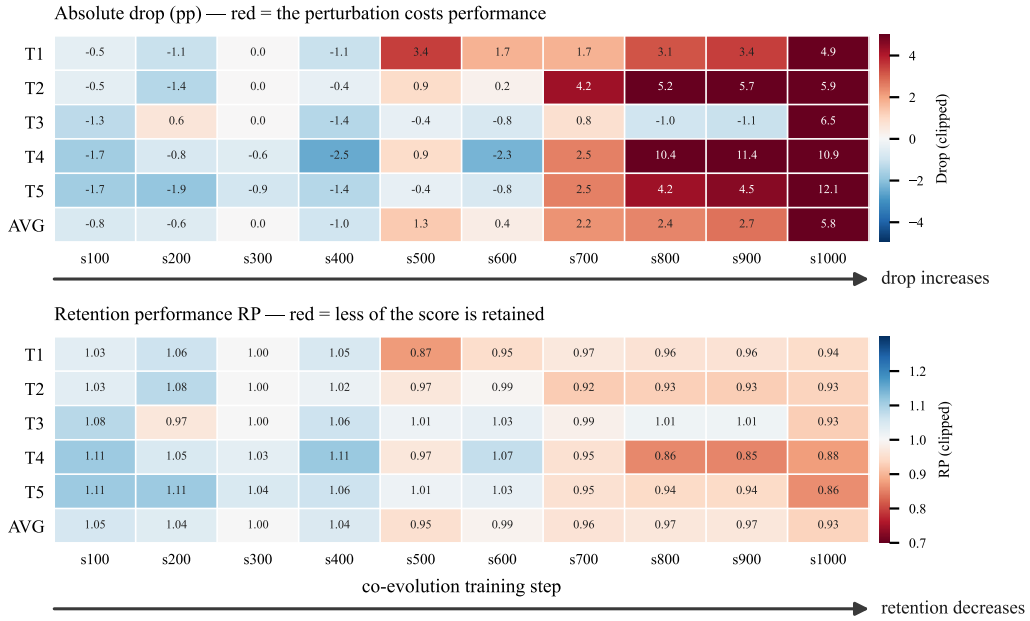


Figure 10: Perturbation response across all checkpoints, ARISE on AMC23. Panels and conventions as in Figure 5; cell values are derived from the per-anchor means in Table 13.

a diagnostic and deliberately keep it out of Table 4: the metric set is closed, and a ratio of two differences is too unstable near a small denominator to carry a pass line.

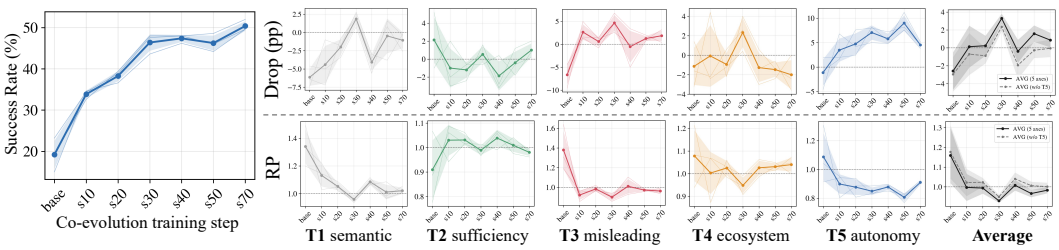


Figure 11: Co-evolution fragility curve for SKILLRL on WebShop; same layout as Figure 4. This is the sole setting in which no content-level drop slope reaches significance.

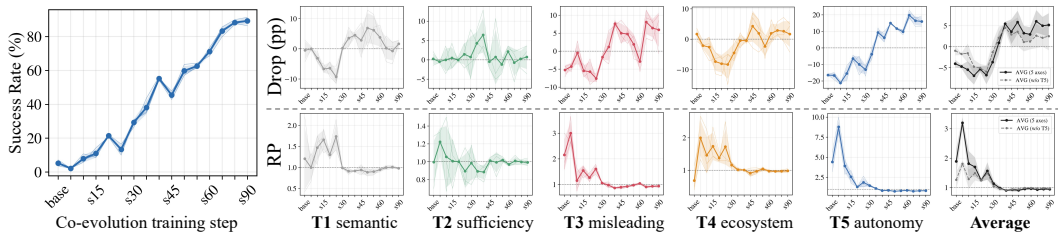


Figure 12: Co-evolution fragility curve for SKILLRL on ALFWorld; same layout as Figure 11. This setting has the widest competence range of the six, from 5.2% to 88.8% over sixteen anchors, so the transition is easiest to read here. Negative drops and RP > 1 at the earliest anchors mean the perturbation *improved* performance, which is admissible for a policy that has not yet co-evolved to the bank (Proposition 1, third clause) rather than a violation of the one-sided prediction. Error bars span ± 1 s.d. over three evaluation repeats that share the policy checkpoint and the same fixed set of rewritten banks and differ only in the rollout sampling seed (temperature 0.4); because each setting comes from a single co-evolution run, these bands quantify evaluation noise and say nothing about variability across training seeds.

An observation, not a prediction. Injected skill text occupies an estimated 15% to 40% of the prompt across the four SKILLRL/RESKILL settings included in this diagnostic, and those with the larger share are *not* systematically the ones with the higher worst-case rates. We record this because it is the first thing a reader will ask about and the measurement is nearly free, but it forms no part of the theory of Section 2 and we make no claim of a relationship.

H.6 T1 MEANING-PRESERVATION QUALITY ASSURANCE

The T1 axis is defined as a meaning-preserving rewrite: the skill's wording changes but its procedural content should not. Because the rewriting suite uses a language model rather than a deterministic operator, we verify this property with an independent, blinded audit whose design specifically defends against the two failure modes a reviewer would flag: the judge being biased toward the rewriter (*self-endorsement*) and the judge being insensitive (a *rubber-stamp*).

Protocol. We audit *all* 190 T1 rewrites in the four SKILLRL/RESKILL settings (91 on SKILLRL-ALFWorld, 84 on SKILLRL-WebShop, 7 on RESKILL-ScienceWorld, 8 on RESKILL-ALFWorld), so there is no sampling. Every pair is scored on a four-level rubric—*equivalent*, *minor drift*, *meaning altered*, *reversed/unrelated*—by two independent judges from different providers: `qwen3-max` (Alibaba) as the primary judge, cross-family with the rewriter (GPT-4o), and `gpt-4o-2024-08-06` (OpenAI) as the secondary judge. The audit is blinded: pair order is shuffled, the original is randomly placed on the A or B side, and axis labels are kept only in an answer key. A human annotator re-checks 50% of the T1 pairs under the same blind protocol.

Judge validity: positive and negative controls. To confirm that the judges are neither over-eager nor over-lenient, the same blinded batch includes 20 identity pairs (a skill against itself) and 40 neg-

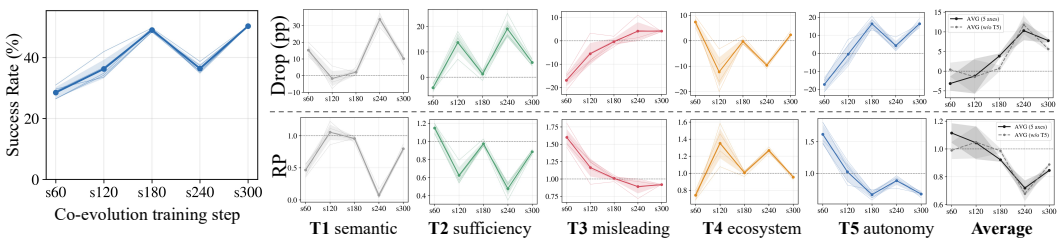


Figure 13: Co-evolution fragility curve for RESKILL on ScienceWorld; same layout as Figure 4. Error bars span five evaluation repeats, and this setting has the fewest anchors of the six (five), yielding the widest slope intervals.

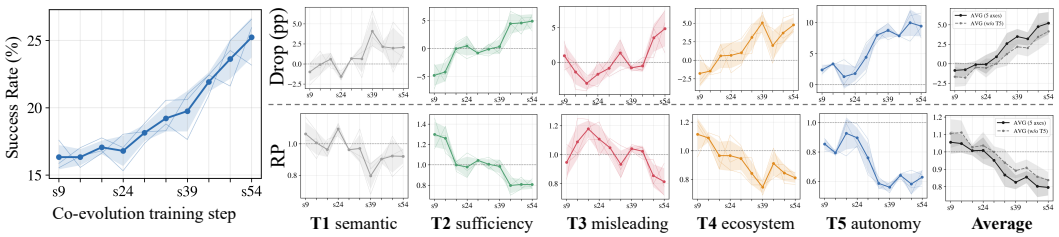


Figure 14: Co-evolution fragility curve for EVOLVING-RL on Mind2Web; same layout as Figure 4.

ative controls (a skill against its T2 or T3 rewrite, which by construction alter the prescribed action). Both judges rate all 20/20 identity pairs *equivalent* and all 40/40 negative controls as non-equivalent (mean rating ≤ 0.36 vs. 1.00 on identity), so they are not rubber stamps and not oversensitive.

Results. Under this test both judges rate *100%* of the 190 T1 pairs as meaning-equivalent ($0/190$ altered; 95% Wilson CI $[0.0\%, 1.9\%]$), with perfect binary agreement between judges (Cohen's $\kappa = 1.0$) and a four-level agreement of 95.8% (all disagreements fall within the negative controls, over the *degree* of alteration, never over its direction). The human re-check on 95 of the 190 T1 pairs (50%) concurs, with no meaning changes found. In these four settings, T1 rewrites therefore preserve meaning, and the corresponding consistency drop in Section 4.4 measures sensitivity to wording rather than rewrite noise.

Skill-bank statistics. Table 14 summarizes the scale and growth mechanism of the skill banks for the four SKILLRL/RESKILL settings. SKILLRL starts from a fixed set of human-authored seed skills and accumulates new skills during training (monotone growth); RESKILL builds from scratch under a fixed library cap.

Table 14: Skill-bank configuration and size for the four SKILLRL/RESKILL settings.

Method	Environment	Mechanism	Initial	Delivered
SKILLRL	ALFWorld	seed + accumulate	55	91
SKILLRL	WebShop	seed + accumulate	66	84
RESKILL	ScienceWorld	from scratch, capped	0	7
RESKILL	ALFWorld	from scratch, capped	0	8

H.7 THE COST IN STANDALONE ABILITY

If co-evolution builds dependence, the natural question is what that dependence is bought with. Figure 17 trains a pure-RL policy on SKILLRL-ALFWorld with the skill mechanism disabled throughout (arm A, no bank at any step) and reads it against the co-evolved delivered checkpoint of the same

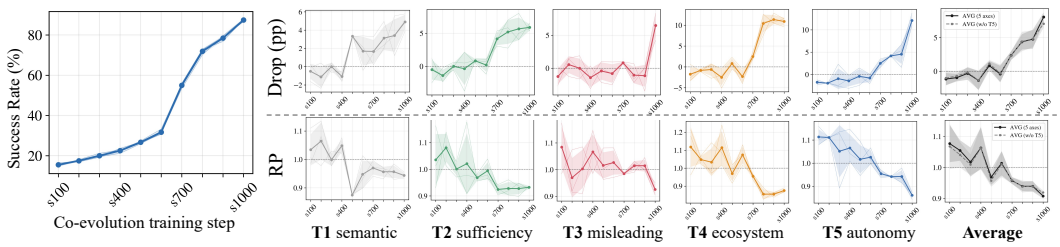


Figure 15: Co-evolution fragility curve for ARISE on AMC23; same layout as Figure 4.

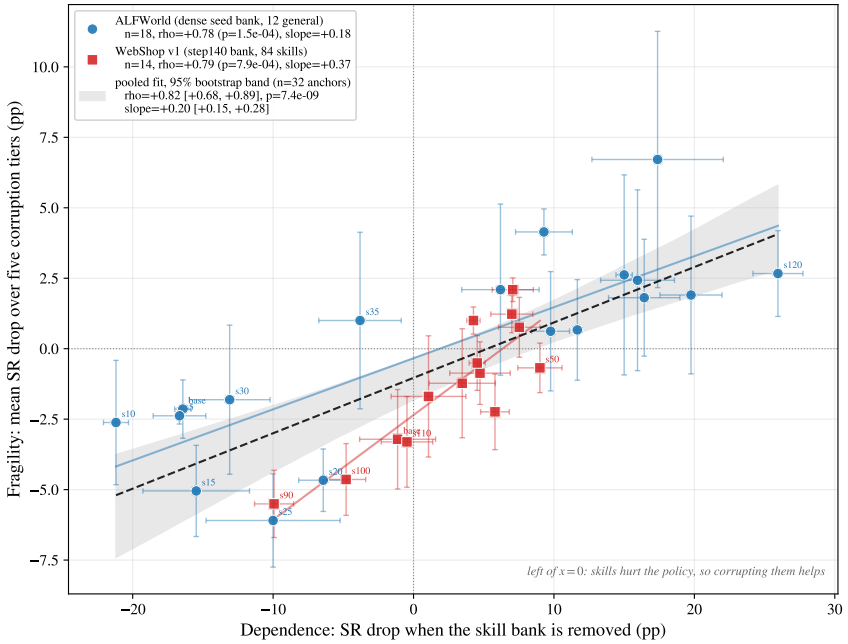


Figure 16: Fragility scales with acquired dependence. Each point is one co-evolution checkpoint (SKILLRL-ALFWorld, circles; SKILLRL-WebShop, squares); error bars are ± 1 s.d. of the same contrast over the three evaluation repeats. x: dependence, the SR drop when the skill bank is removed. y: fragility, the mean SR drop over the five corruption tiers. Solid lines are per-setting fits; the dashed line and grey band are the pooled fit with its 95% bootstrap envelope over anchors. Points left of $x = 0$ are checkpoints that have not yet co-evolved to the bank, where removing or corrupting skills improves performance.

setting, both with the bank present (84.1%, $n = 5$) and with it removed (57.6%, $n = 5$; step-115 bank, evaluated on the `eval_in_distribution` split at temperature 0.4). Two things stand out. First, the co-evolved policy stripped of its bank scores 57.6%, whereas a policy that never trained with skills reaches a last-five-checkpoint mean of 70.6% and a peak of 87.5%: co-evolution does not merely add a removable crutch, it leaves the underlying policy *worse at acting alone* than one that was never given the crutch. The dependence measured in Appendix H.4 is therefore not free—it is standalone ability traded away. Second, the pure-RL run is itself unstable: it peaks at 87.5% near step 125 and then falls to 53.1% by step 150, with entropy and KL rising together over the same window (panels B, C). Neither training regime yields a monotone capability curve; the distinctive cost of co-evolution is the additional erosion of the skill-free policy.

The comparison is between two evaluations that share the `eval_in_distribution` split and the temperature-0.4 sampling but differ in sample count (the training-time validation draws 64 tasks, the CODA reference 140) and in the ALFWorld backend (textual `AlfredTWEnv` during training

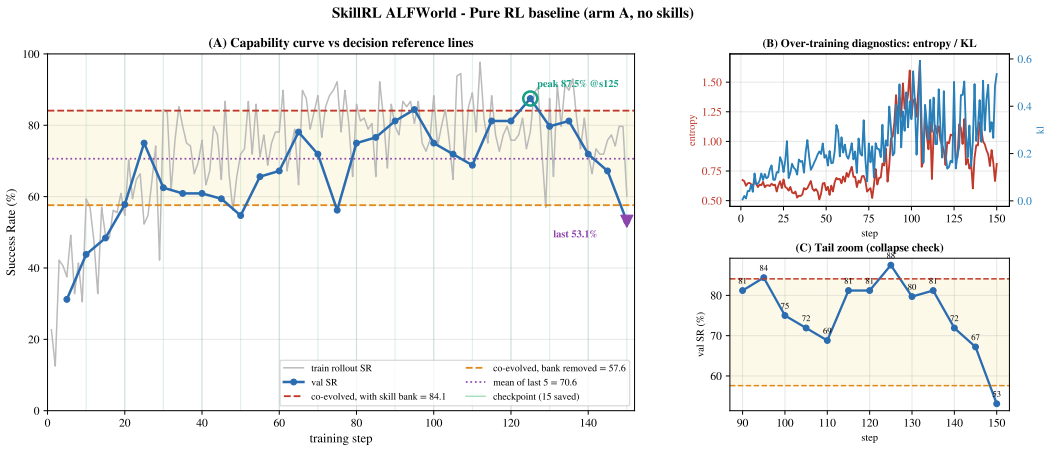


Figure 17: Pure-RL baseline (arm A, skills disabled) on SKILLRL-ALFWorld. (A) Validation SR over training, against the co-evolved delivered checkpoint with the bank present (red, 84.1%) and removed (orange, 57.6%); the co-evolved policy with its bank removed sits below even the pure-RL last-five mean (purple dotted, 70.6%). (B) Entropy and KL rise together in the second half, the signature of an over-training collapse. (C) Tail zoom: the run falls from its step-125 peak to 53.1% at step 150. Both curves and reference lines use the `eval_in_distribution` split at temperature 0.4; see the text for the difference in sample count and backend.

versus the simulator-backed `AlfredThorEnv` at evaluation); the reference lines should be read as the level of the co-evolved policy on the shared split, not as a point comparison to two decimal places. We report this as a supporting observation rather than a headline result.