SKILL LOCK-IN: SILENT RETIREMENT FAILURES IN SELF-EVOLVING LLM SKILL LIBRARIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Self-evolving LLM agents can retain harmful advice in growing skill libraries despite continued evaluation. Judge bias affects skill removal and admission differently. When feedback weakens measured effects without changing their sign, zero remains a meaningful decision boundary, while a permissive admission rule can allow much greater true harm. We identify sparse skill use, optimistic feedback and routing changes as interacting causes of this *skill lock-in*. We introduce PAWL (Paired Anytime-valid Watchdog for Libraries), a lifecycle that estimates a skill's contribution on the tasks where it is used. It randomly withholds skills to maintain task-matched no-skill references and uses confidence bounds valid under repeated inspection to govern removal and admission. Experiments across four solver families and two live LLM judges show stronger harmful-skill detection and more selective admission. In one matched optimistic-feedback setting, PAWL retires 16 of 18 planted harmful skills, versus two for the success-count baseline. Separate replays show that normalizing success counts by skill use remains insufficient. On screened, loop-synthesized coding skills, PAWL admits 15 of 18 helpful candidates and none of 18 harmful candidates under strong simulated optimistic feedback within the tested budget. These findings establish measurement, threshold choice and sustained exposure as central design considerations for self-evolving skill libraries.

1 INTRODUCTION

Self-improving agents turn experience into reusable skills, reflections and workflows (Wang et al., 2023; Shinn et al., 2023; Zhao et al., 2024; Chen et al., 2024; Wang et al., 2024b). Continual skill creation makes deciding what to admit, retain and retire an ongoing requirement (Li, 2026). We study natural-language skills supplied to a solver. Incorrect guidance can repeatedly misdirect the solver and propagate into later skills (Shang et al., 2026; Shao et al., 2026). As libraries grow, more skills compete for limited evaluation, making curation central to preserving the benefits of accumulated experience.

Recent systems validate, gate and bound the library (Yang et al., 2026; Moll et al., 2026; Shen et al., 2026), but an imperfect LLM judge can mislabel the failures that should trigger removal. Harmful skills can then persist despite continued evaluation and revision, a failure we call *skill lock-in*. A router selects skills that seem applicable to each task, so raw success mixes task difficulty with skill effect. Averaging over all tasks dilutes the signal from sparse use; judge errors can weaken or shift it. Routing changes can then stop sampling an unresolved skill. Our experiments separate these failures: normalizing by skill use leaves judge distortion unresolved, while preserving routed exposure can recover stalled retirement decisions.

We introduce PAWL (**P**aired **A**nytime-valid **W**atchdog for **L**ibraries). For each routed task, the engine randomly withholds the skill for an audit solve and maintains a task-matched no-skill reference (Figure 1). The residual $d'_i = y'_i - \widehat{p}_0(x_i)$ subtracts estimated no-skill success $\widehat{p}_0(x_i)$ from the reported binary outcome y'_i (pass or fail) on task x_i at skill observation i. Throughout, primes distinguish reported feedback from true outcomes. Residuals enter an anytime-valid confidence sequence: an interval that remains valid across repeated inspections. Newly proposed skills, or candidates, enter the *shadow* state for probation, a trial period on routed tasks. Admission moves them into the *active* library for routine use. A negative upper bound certifies retirement; *retired* skills

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.

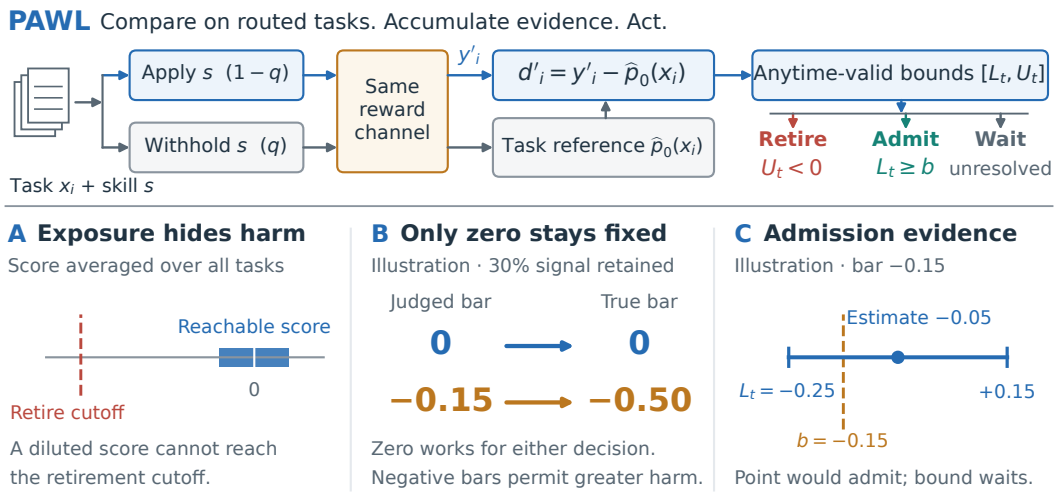


Figure 1: **PAWL lifecycle and design principles.** Top: auditing and decisions (q: audit probability; $[L_t, U_t]$: bounds on the weighted residual mean after t skill observations; b: admission bar). Bottom: illustrations of exposure dilution (A), threshold distortion (B), and admission uncertainty (C). Panel B assumes constant arm-blind feedback and a matched population reference.

are excluded from further routing. Undecided candidates remain in probation until their evidence budget expires (Figure 1C). Anytime validity also permits evidence racing: adaptively choosing which candidate to sample next.

The choice of threshold is equally consequential. An arm-blind judge has the same error rates with and without a skill. When those rates are constant and leave positive signal, the population skill-minus-no-skill effect shrinks toward zero. Zero remains a meaningful boundary for both retirement and admission, but a permissive negative admission bar allows more true harm as feedback worsens. This gives a practical design principle: anchor removal at zero and choose the admission tolerance explicitly, accounting for its evidence cost. PAWL's finite-reference implementation and the assumptions connecting its residuals to this population result are specified in Section 4.

Our experiments span coding and instruction-following suites, multiple solver families, simulated corruption and live judges. The contributions are:

- **Diagnosis and threshold principle.** We separate exposure dilution, judge distortion and routing displacement as causes of lock-in. We show why zero preserves its meaning under informative arm-blind feedback, while a negative admission threshold tolerates increasing harm as feedback worsens.
- **Curation from routed comparisons.** PAWL uses randomized no-skill audits and anytime-valid bounds for retirement and admission, allowing adaptive evidence allocation.
- **Harm detection and useful admission.** Matched experiments recover retirement under biased feedback. On screened synthesized coding skills, PAWL admits helpful candidates while filtering harmful ones accepted by a rule that thresholds the paired mean. Allocation studies show when continued exposure restores decisions and how task reach limits utility.

2 Related Work

Self-evolving skill libraries. Recent work treats reusable guidance as a managed library (Li, 2026). Prior work documents drift under unfiltered accumulation, uses success counts to retire poorly performing skills, and shows that optimistic feedback can disable retirement despite stable aggregate quality (Zhang et al., 2026a;b;c). SkillOpt and GRASP gate edits through held-out improvement and regression checks using benchmark correctness feedback (Yang et al., 2026; Moll et al., 2026), while Verifier-as-Gatekeeper screens skills and their combinations before use (Shang et al., 2026).

Matched rollout groups assess candidates before storage (Zhang et al., 2026d), and leave-one-skill-out validation guides retirement (Shen et al., 2026). SkillAudit uses paired trajectories to guide edits, with a fixed task-derived verifier to veto regressions (Gao et al., 2026); BACKTRACE distinguishes stated use from intervention-measured influence (Hu et al., 2026). This work separates measurement failures from loss of routed evidence, analyzes how judge errors change the meaning of lifecycle thresholds, and tests the resulting design choices through harm detection and useful-skill admission.

Sequential inference and LLM judges. Confidence sequences support valid inference under optional stopping and adaptive continuation (Howard et al., 2021; Waudby-Smith & Ramdas, 2024); recent work applies anytime-valid gates to steering adapters and execution harnesses around a frozen model (Sengupta, 2026). LLM-judge studies identify positional bias (Wang et al., 2024a), self-preference (Panickssery et al., 2024), and reasoning limitations (Zheng et al., 2023).

3 Diagnosing Skill Lock-In Under a Noisy Judge

Exposure dilution. Consider a retirement statistic that averages bounded per-use scores over all task opportunities. For skill s, let $z_{s,i} \in [-1, 1]$ be the score on use i (lower is worse), n_s its number of uses, and N_s all tasks since creation. With exposure fraction $a_s = n_s/N_s$ and mean per-use score $\bar{z}_s$, the globally averaged statistic is

$$G_s = \frac{1}{N_s} \sum_{i=1}^{n_s} z_{s,i} = a_s \bar{z}_s \in [-a_s, a_s]. \tag{1}$$

A fixed negative cutoff $-\gamma$, with $\gamma > 0$, requires a margin of unfavorable scores before retirement (Zhang et al., 2026c). Yet the statistic cannot reach this cutoff when $a_s < \gamma$, regardless of the observed scores (Figure 1A). Normalizing by uses removes this dilution, but the per-use score must still measure the skill's contribution.

Routed evidence. Let a frozen solver receive task x and optionally one natural-language skill s. Skills compete for routing but are not composed within a solve. A router either selects s or abstains. The lifecycle targets marginal contribution $c(s)$ on the distribution of tasks routed to the skill,

$$c(s) = \mathbb{E}_{x \sim \mathcal{D}_s} \left[y(x; s) - y(x; \varnothing) \right], \tag{2}$$

where $y \in \{0, 1\}$ denotes true correctness, $\varnothing$ means no skill, and $\mathcal{D}_s$ is the routed task distribution. Replacing y with the reported label y' defines the judged contribution $c'(s)$. Raw success on $\mathcal{D}_s$ mixes task difficulty with skill influence. PAWL compares a skill-arm outcome with a task-specific no-skill reference $\widehat{p}_0(x)$, maintained by randomized audits and router abstentions. The reference is predictable: fixed before the skill-arm outcome is observed. For skill observation i, reported pass/fail feedback $y'_i \in \{0, 1\}$ contributes

$$d'_i = y'_i - \widehat{p}_0(x_i) \in [-1, 1]. \tag{3}$$

The residual's scale does not shrink with exposure; Section 4 clarifies how reference estimation affects interpretation of Equation (2). Even with this measurement, competing skills can displace an unresolved skill before its interval resolves. Continued exposure is therefore needed to turn routed comparisons into lifecycle decisions.

Judge distortion. Labeling failures as successes can shift an absolute performance score away from removal (Zhang et al., 2026a). Persistent bias does not average away with more observations. Relative comparisons also depend on how the judge treats skill and reference outputs. For population arm means, let d and d' denote true and judged residuals formed with matched population no-skill references. An informative binary judge with constant error rates then satisfies

$$\mathbb{E}[d'] = \kappa \, \mathbb{E}[d] + \Delta_{\text{eff}}, \qquad \kappa > 0, \tag{4}$$

where κ is the signal remaining after attenuation and Δ_{eff} is the effective skill-versus-reference offset. With constant arm-blind error rates, $\Delta_{\text{eff}} = 0$: the additive offset cancels, while attenuation remains.

Under constant informative arm-blind feedback and matched population references, valid confidence bounds translate this relation into decisions: an upper bound below zero certifies a negative weighted

true contribution; a lower bound at zero certifies non-negativity. A negative admission bar b instead certifies only the true-contribution lower bound $\max(-1, b/\kappa)$, allowing greater harm as the judge's signal κ weakens (Figure 1B). Appendix A.3 gives the channel derivation and formal guarantees (Propositions 1 and 2).

4 PAWL: PAIRED, ANYTIME-VALID CURATION

PAWL turns the diagnosis into three linked design choices: collect comparison evidence on routed tasks, accumulate it with time-uniform inference, and allocate further exposure to unresolved candidates. The lifecycle uses established confidence-sequence tools (Howard et al., 2021; Waudby-Smith & Ramdas, 2024); its design connects these tools to the measurement and threshold requirements above.

Operation. Algorithm 1 collects routed evidence and updates the lifecycle at round end. Task references start from clean executable labels, phased out by the default cold-start policy as in-run no-skill evidence accumulates. Audit attempts and router abstentions update these shared references; only skill-arm outcomes with a pre-outcome reference enter the residual sequence.

Decision rules. For each skill, PAWL maintains an anytime-valid empirical-Bernstein confidence sequence $[L_t, U_t]$ that adapts to observed variation and is simultaneous over skills and decision times (Appendix A). Certified retirement takes priority over admission. Budget expiry removes unresolved candidates without asserting harm. Admission margins are design choices: $b = -\tau_{\text{adm}}$ with tolerance $\tau_{\text{adm}} = 0.15$ by default, or $b = +0.02$ for stricter positive-utility selection. Both thresholds concern the residual mean.

Algorithm 1 PAWL: evidence collection and round-end decisions.

Require: active library $\mathcal{B}$, shadow cohort $\mathcal{Q}$, audit rate q, bar b, attempt budget B; $n(s)$ counts skill-arm attempts for s

for each round **do**
  prioritize applicable shadow skills toward their current attempt quotas
  for each training attempt on task x **do**
    $s \leftarrow \text{ROUTE}(x, \mathcal{B}, \mathcal{Q})$
    if $s = \varnothing$ or a randomized audit is drawn with probability q **then**
      solve without a skill; update the reference $\widehat{p}_0(x)$
    else
      solve with s; observe y'; increment $n(s)$
      append $d' = y' - \widehat{p}_0(x)$ if the reference exists
    end if
  end for
  for each active or shadow skill s **do**
    compute confidence bounds $[L(s), U(s)]$ from its ordered residual sequence
    if $U(s) < 0$ **then**
      retire s relative to the estimated reference
    else if $s \in \mathcal{Q}$ and $L(s) \geq b$ **then**
      activate s
    else if $s \in \mathcal{Q}$ and $n(s) \geq B$ **then**
      expire s
    end if
  end for
  optionally race shadow skills and reallocate the next evidence quota
  fill free shadow slots from queue with zero counts and empty residual sequences
end for

Certificate target. Writing c'_i for the conditional effect on reported feedback and $p'_0(x_i)$ for the corresponding no-skill rate, each residual has conditional mean $\mu'_i = c'_i + p'_0(x_i) - \widehat{p}_0(x_i)$. The confidence sequence covers a predictable weighted average of the μ'_i. "Certified retirement" therefore refers to a negative residual mean; causal interpretation additionally requires the judge-channel

assumptions above and control of reference error, which the deployed interval does not include. A randomized two-arm alternative and empirical reference checks are given in Appendices A.1 and C.3.

Evidence allocation. Anytime validity permits adaptive continuation. Optional successive-halving racing (Karnin et al., 2013; Jamieson & Talwalkar, 2016) ranks shadow candidates by their observed residual means after a 12-attempt warm-up. It permanently removes the bottom half without claiming harm and gives survivors up to three times a fair quota. The same confidence bounds govern admission.

5 Experimental Setup

Research questions (RQs). We ask whether routed comparison recovers retirement under simulated corruption (RQ1), how live judges and admission rules affect lifecycle decisions (RQ2), and how evidence allocation translates into certification and held-out utility (RQ3).

Tasks and models. MBPP+ is the principal coding testbed (Austin et al., 2021; Liu et al., 2023); IFBench supplies verifiable instruction following (Pyatkin et al., 2025); BIRD-SQLite tests database queries scored by execution (Li et al., 2023); HumanEval+ and BigCodeBench-Hard test frozen-library transfer (Chen et al., 2021; Liu et al., 2023; Zhuo et al., 2025). The main solver and skill synthesizer, which proposes guidance from failures, are Claude Opus 4.7, with Kimi K2.5, GLM-5 and GPT-5.5 replications; GLM-5 also serves as a second live judge. Primary MBPP+ comparisons use three random seeds, extended to six on the two false-pass channels; Appendix B specifies configurations and evaluation controls.

Comparisons. We compare no skills, append-only accumulation, the success-count lifecycle of Zhang et al. (2026c), PAWL, a point-estimate admission ablation that thresholds the same residual mean, and an adapted SkillOpt/GRASP-style held-out regression gate (Gate) (Yang et al., 2026; Moll et al., 2026). Here the gate reads judged labels and forces skills onto reserved validation tasks; a clean held-out evaluator would avoid the judge-channel distortion. Within each study, lifecycle arms share the solver, router and synthesis procedure. Recorded-sequence replays separately test use normalization, fixed-budget pairing and alternative confidence bounds.

Interventions. Canaries are planted skills labelled by an offline screen before insertion. Designed harms give plausible but damaging guidance; helpful and near-neutral controls probe unnecessary removal. A separate admission study screens *natural* skills synthesized in earlier runs. MBPP+ pairs six helpful with six harmful natural skills; IFBench pairs six helpful natural skills with three designed formatting harms. Appendix B.4 gives instructions, screening uncertainty and selection criteria.

Feedback. Clean feedback uses executable grades. Simulated false pass at rate r flips failing labels to pass with probability r; false fail does the reverse, and symmetric noise flips either label. Live judges supply training labels for named harm (explicit in the prompt) or omitted harm (inferred from the task). Unanswered judge calls fall back to executable labels; the critic and synthesizer retain executable error traces.

Held-out evaluation. Frozen libraries are compared task-by-task with no-skill solving at the same sampling budget. All MBPP+ conditions supply the required entry-point name; IFBench routing uses instruction-family metadata. Of the 200 held-out IFBench tasks, 105 use instruction types absent from the 60-task training split, testing whether skills apply beyond the training distribution.

Decision audits and evaluation cohorts. Executable audits provide descriptive checks on the skill's routed task set; their task-averaged intervals do not establish causal error control at adaptive decision times. Primary retirement comparisons include every matched plant, keyed by channel, seed and skill, and score both arms at their earlier run end. A sensitivity analysis requires at least 20 rounds after insertion in both arms, regardless of outcome. Admission reports all planted candidates, including unresolved ones; a "below-bar admission" has pre-insertion screened effect below -0.15, the chosen tolerance.

6 RESULTS

6.1 RQ1: PAIRING RECOVERS RETIREMENT UNDER BIASED FEEDBACK

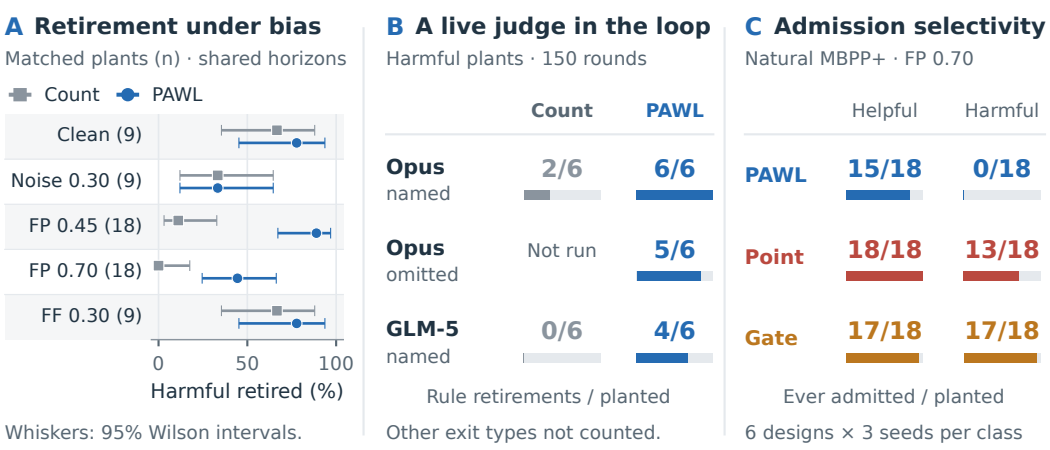


Figure 2: **Retirement and admission under imperfect feedback.** (A) All matched harmful plants at 60 rounds (FP/FF: false pass/fail). Whiskers are 95% Wilson intervals. (B) Rule-triggered retirement under live judges with auxiliary executable feedback. (C) Admission of screened natural MBPP+ skills under simulated bias, with expanded probation over 150 rounds; admissions include later removals. Gate: adapted SkillOpt/GRASP-style held-out diagnostic.

At false-pass 0.45, PAWL retires $16/18$ harmful plants against $2/18$ for the success-count rule; at 0.70, the outcomes are $8/18$ and $0/18$ (Figure 2A). Paired seed-cluster bootstrap confidence intervals (CIs) for the detection advantage are $[0.56, 0.94]$ and $[0.22, 0.61]$ across six seeds. The advantage survives requiring at least 20 rounds of follow-up independently of outcome (Appendix Table 4).

Repairing the count baseline. On the same recorded MBPP+ and IFBench observations, PAWL detects more harmful plants and retires fewer screen-labelled non-harmful controls than use-normalized counting with a 20-use warm-up (Table 1). At false-pass 0.70, normalized counting detects $0/31$ harmful sequences versus $14/31$ for PAWL: removing dilution does not remove optimism in raw success. Fixed-budget pairing detects more harm among sequences reaching each tested budget; earlier PAWL retirements are excluded. These conditional replays do not rank overall detection efficiency. Time-uniform inference permits flexible stopping and continuation.

Table 2 extends the comparison to three additional solvers: PAWL detects more harmful plants without retiring their screen-labelled non-harmful controls. In the primary MBPP+ study, control retirements decrease under false-fail feedback but increase under clean feedback (Appendix C.1); specificity remains dependent on the setting.

Among primary MBPP+ plants retired by PAWL, median attempts increase from 48 on clean feedback to 138 at false-pass 0.45 and 332 at 0.70, qualitatively matching the $1/\kappa^2$ cost of estimating an attenuated effect. On IFBench, the rules separate under clean feedback, but neither resolves harm under strong false-pass bias (Table 2). Newer skills can displace unresolved plants before enough evidence arrives. An exposure floor reserves further routed attempts for unresolved skills; Section 6.3 examines the allocation tradeoff.

Reference reuse and confidence bounds. On the same recorded sequences at the default audit rate, restricting task references to prior randomized audits reduces harmful-skill retirements from $66/98$ to $24/98$. The tested randomized two-arm alternative retires $1/98$ (Appendix Table 6). Reference reuse supports detection within the tested budget.

A valid stitched Hoeffding bound detects more harmful plants sooner in replay, without more screen-labelled non-harmful retirements (Appendix Table 5A). Its clean-feedback advantage in full-loop detection reverses under strong false-pass bias (Table 5B). Retirement changes subsequent routing, so a fixed-sequence ranking need not predict an adaptive lifecycle.

Table 1: **Denominator repair and fixed-budget pairing.** Retired/available MBPP+ and IFBench sequences; n counts observations and non-harmful denotes screened controls.

A. Use normalization (all prefixes)			
Evidence	**Rule**	**Harmful**	**Non-harmful**
All prefixes	Use-normalized count, no warm-up	$74/98$	$68/93$
All prefixes	Use-normalized count, $n \geq 20$	$45/98$	$38/93$
All prefixes	PAWL	$66/98$	$6/93$
B. Fixed budgets (earlier PAWL retirements excluded)			
$n = 20$	Fixed-budget paired test	$15/86$	$0/91$
$n = 20$	PAWL	$2/86$	$0/91$
$n = 60$	Fixed-budget paired test	$14/62$	$3/87$
$n = 60$	PAWL	$3/62$	$0/87$
$n = 180$	Fixed-budget paired test	$17/33$	$1/27$
$n = 180$	PAWL	$4/33$	$0/27$

Early retirements excluded. At $n = 180$, 41 harmful and 5 control sequences ended after PAWL retirement. Including them gives $45/98$ harmful and $5/93$ control crossings by that budget. Alternative fixed-budget outcomes for these shortened sequences are unknown.

Finding 1. Pairing plus an anytime-valid bound restores harmful-skill retirement where success counts fail. Under the tested simulated channels, optimistic feedback primarily raises the evidence cost, while routing displacement can prevent that evidence from arriving. On IFBench at false-pass 0.70, an exposure floor retires the first harmful plant in $6/6$ runs across two 200-round follow-up studies (Appendix C.7).

Table 2: **Retirement beyond the primary MBPP+ setting.** Entries are retirements/all matched plants at shared horizons. Non-harmful denotes screen-labelled controls; dashes indicate no corresponding plants.

Suite	**Solver**	**Reward channel**	**Lifecycle**	**Harmful**	**Non-harmful**
IFBench	Opus 4.7	Clean	Success count	$0/9$	–
IFBench	Opus 4.7	Clean	PAWL	$5/9$	–
IFBench	Opus 4.7	False pass 0.70	Success count	$0/9$	–
IFBench	Opus 4.7	False pass 0.70	PAWL	$0/9$	–
MBPP+	Kimi K2.5	Clean	Success count	$4/6$	$3/6$
MBPP+	Kimi K2.5	Clean	PAWL	$5/6$	$0/6$
MBPP+	GLM-5	Clean	Success count	$1/6$	$1/6$
MBPP+	GLM-5	Clean	PAWL	$6/6$	$0/6$
MBPP+	GPT-5.5	Clean	Success count	$3/6$	$1/6$
MBPP+	GPT-5.5	Clean	PAWL	$6/6$	$0/6$

6.2 RQ2: LIVE-JUDGE RETIREMENT AND USEFUL-SKILL ADMISSION

For planted harm, two live judges supplied training labels over 150 rounds: Opus 4.7 judging its own solver on named and omitted harm, and GLM-5 judging the same solver on named harm (Figure 2B). Across the Opus planted-harm runs, judged training success is about 0.87, against 0.59 from the executable grader. Yet relative comparisons retire more named-harm plants than the count rule under both judges and retain signal even when the prompt omits the harm. With the Opus judge, the count baseline also loses three named-harm plants to capacity eviction; its remaining plant is never used. One omitted-harm PAWL plant exits through a duplicate merge.

For GLM-5, the two plants PAWL leaves active have judged contributions within 0.04 of zero against task-averaged executable effect of -0.62. Of the 23 certified Opus-judge planted-harm retirements, including variants that floor and cap routed attempts, 21 have executable intervals below zero; all four GLM-judge retirements do too. The remaining two Opus cases are unresolved by these descriptive audits (Appendix C.3).

Admitting useful skills. Expanded probation supports useful admission of natural MBPP+ skills, with capacity, insertion timing and horizon changed together to give every plant a slot. PAWL admits all helpful candidates under clean feedback; under simulated false-pass bias, helpful admissions rise from $3/18$ in the default cohort to $15/18$, with no screened harmful admissions in either cohort or channel (Figure 2C; Appendix Table 8). All 15 helpful admissions under bias use only in-run judged references, without clean initial labels.

The paired point-estimate ablation admits $13/18$ harmful candidates in the expanded biased cohort. PAWL expires 17 harmful candidates and leaves one unresolved, reflecting its evidence requirement and probation budget. Each of the six helpful designs is admitted in at least two of three runs under bias. Retrospectively restricting the recorded trajectories to the four designs with screened effects at least $+0.30$ retains $12/12$ helpful admissions on clean feedback and $11/12$ under bias.

The adapted gate exposes allocation and measurement failures: under clean MBPP+ feedback, 17 of 18 helpful plants never receive a reading because the scheduler prioritizes the most-used skills. Biased runs have fewer judged failures and synthesized competitors, and every plant is read. Nearly all are admitted, then all admitted harms and most admitted helpful skills are retired. Its reserved tasks ignore applicability and require extra solves, precluding a matched-budget ranking. The bound also has a measurable cost: on IFBench under bias, it admits only $3/18$ helpful candidates while twelve exhaust probation.

Judge distortion and horizon. Average optimism does not imply arm blindness. On planted harm, the judged skill-minus-reference effect is more negative than the executable effect, whereas arm-blind optimism would move it toward zero. This relative penalty appears for both named and omitted harm and varies across skills. Figure 3A varies the skill-minus-reference false-pass offset Δ_0, equal to Δ_{eff} under equal-arm attenuation. Negative offsets raise retirement rates among skills whose executable intervals do not establish harm. These rates serve as an audit proxy. The measured judges have unequal, sometimes negative attenuations, so their offsets alone do not establish a safety margin.

A long-horizon check under the live Opus judge and an exposure floor retires $12/18$ screened near-neutral plants over 1,000 rounds. Among these retirements, eleven executable audit intervals include zero and one lies below it (Appendix C.6). This reveals an exposure tradeoff: continued sampling resolves displacement, but can also retire skills whose harm remains unconfirmed.

Finding 2. Routed comparisons retain enough signal to detect planted harm under the tested live judges. On screened natural coding skills with simulated bias, PAWL accepts helpful candidates while filtering harmful ones. Evidence requirements limit useful acceptance where exposure is scarce.

6.3 RQ3: EVIDENCE ALLOCATION AND HELD-OUT UTILITY

Sustaining exposure. In the MBPP+ probation study, candidates receive a median one observation per round against a quota of three. In a separate experiment, flooring and capping unresolved active skills' attempts reduces the median post-planting development-score drop from 0.380 to 0.140, but slows retirement without improving overall development scores (Appendix C.7). Evidence allocation must therefore balance resolving existing skills with testing promising candidates.

Positive-utility selection and racing. In recorded-sequence replays, the tested bound makes fewer positive-utility selections than the point rule, with a higher median evidence cost (Figure 3B). At the point rule's first crossings, the median lower bound is still -0.584: a positive estimate need not yet support a positive-effect claim. In full-loop IFBench studies, racing tests more candidates than round robin (equal quotas assigned in turn) at both thresholds and increases certifications at the positive-utility bar (Figure 3C). At the negative bar, ranking also discards candidates whose point estimates already exceed that permissive threshold, and admissions decrease.

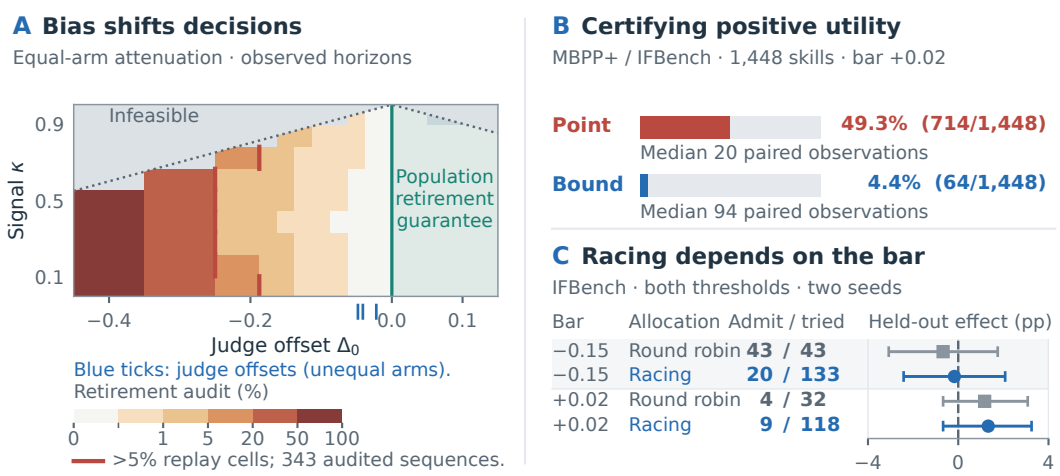


Figure 3: **Judge distortion and evidence allocation.** (A) Equal-arm replay: red marks audit-proxy retirement rates above 5%; green marks the population guarantee under constant informative channels and a matched population reference. Blue ticks show unequal-arm judge offsets only. (B) Positive-utility selection on recorded sequences. (C) Racing throughput and held-out effects with 95% task-seed bootstrap intervals; pp denotes percentage points.

Utility and task reach. PAWL maintains compact libraries across suites, while held-out intervals do not establish a broad accuracy gain (Appendix Table 9). Append-only reduces accuracy on BIRD-SQLite, showing that library growth itself can be harmful. Limited task reach helps explain the gap between certification and library value: IFBench utility-racing libraries engage (route a skill to) only 4.5–9.5% of held-out tasks. Across MBPP-trained libraries, pooled engagement falls from 36.0% on HumanEval+ to 1.3% on BigCodeBench-Hard (Zhuo et al., 2025), although both use Python. This ordering holds under all three tested lifecycles, suggesting that transfer depends on task compatibility as well as library curation.

Finding 3. Racing increases candidate throughput and, at a positive bar, certifications. Task reach is a separate bottleneck: selected skills must apply to held-out tasks for better curation to improve library value.

7 LIMITATIONS AND CONCLUSION

Scope. The judge-channel guarantees assume a matched population reference and a homogeneous informative channel. Persistent offsets from judge distortion or reference error remain as the residual interval narrows; negative offsets can cause retirement without true harm (Appendix C.6). Sustained operation may require calibration. Cumulative evidence can also delay detection when a helpful skill deteriorates.

Independent harm validation at retirement remains limited for synthesized and screened near-neutral skills. Live studies retain auxiliary executable feedback. Designed harms concern interface and formatting changes; natural-skill labels use only one to thirteen screening tasks per skill. Held-out results establish neither accuracy improvement nor equivalence. Reference-aware inference and broader skill coverage remain open directions.

Conclusion. Skill lock-in arises from diluted or distorted measurements and interrupted exposure. PAWL combines routed comparison, time-uniform decisions and explicit admission thresholds to address these failures. Controlled experiments recover retirement under severe false-pass bias and demonstrate selective admission of helpful synthesized coding skills. These findings show why growing libraries need meaningful decision criteria and enough applicable tasks for evaluation.

AI USE STATEMENT

Generative AI tools assisted with coding and writing, including method implementation, experiment execution and auditing, analysis and interpretation of results, and figure preparation. Verification included code tests and checks against experimental records. The authors take responsibility for all content, including AI-assisted text, claims and artifacts.

ETHICS STATEMENT

This work uses public benchmarks, with no human subjects or personal data. Synthetic harmful skills test lifecycle decisions, aiming to improve self-evolving systems under imperfect feedback.

REPRODUCIBILITY STATEMENT

The supplementary material includes the experimental implementation. Appendix A gives statistical assumptions, confidence-bound construction and proofs. Appendix B specifies data splits, model and lifecycle settings, screening procedures and judge prompts. Appendix C reports supporting analyses, including cohort sensitivity, baseline comparisons and decision audits.

REFERENCES

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Minghao Chen, Yihang Li, Yanting Yang, Shiyu Yu, Binbin Lin, and Xiaofei He. AutoManual: Constructing instruction manuals by LLM agents via interactive environmental learning. In *Advances in Neural Information Processing Systems*, volume 37, pp. 589–631, 2024. doi: 10.52202/079017-0019.

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

Steven R. Howard, Aaditya Ramdas, Jon McAuliffe, and Jasjeet Sekhon. Time-uniform, nonparametric, nonasymptotic confidence sequences. *The Annals of Statistics*, 49(2):1055–1080, 2021. doi: 10.1214/20-AOS1991.

Jinwei Hu, Yi Qi, Xinmiao Huang, Youcheng Sun, Yi Dong, and Xiaowei Huang. Skill use or skill theater? evaluating the reasoning backroom in skill-augmented language agents. *arXiv preprint arXiv:2607.27484*, 2026.

Kevin Jamieson and Ameet Talwalkar. Non-stochastic best arm identification and hyperparameter optimization. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pp. 240–248, 2016. URL https://proceedings.mlr.press/v51/jamieson16.html.

Zohar Karnin, Tomer Koren, and Oren Somekh. Almost optimal exploration in multi-armed bandits. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pp. 1238–1246, 2013. URL https://proceedings.mlr.press/v28/karnin13.html.

Jinyang Li, Binyuan Hui, Ge Qu, Jiaxi Yang, Binhua Li, Bowen Li, Bailin Wang, Bowen Qin, Ruiying Geng, Nan Huo, Xuanhe Zhou, Chenhao Ma, Guoliang Li, Kevin C. C. Chang, Fei Huang, Reynold Cheng, and Yongbin Li. Can LLM already serve as a database interface? a BIg bench for large-scale database grounded text-to-SQLs. In *Advances in Neural Information Processing Systems*, volume 36, pp. 42330–42357, 2023. doi: 10.52202/075280-1835.

Yubo Li. Dynamic agent skills: A lifecycle survey and taxonomy of evolving skill libraries. *Transactions on Machine Learning Research*, 2026. URL https://openreview.net/forum?id=cjU3YbcRr8.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by ChatGPT really correct? rigorous evaluation of large language models for code generation. In *Advances in Neural Information Processing Systems*, volume 36, pp. 21558–21572, 2023. doi: 10.52202/075280-0943.

Johannes Moll, Jean-Philippe Corbeil, Jiazhen Pan, Martin Hadamitzky, Daniel Rueckert, Lisa Adams, and Keno Bressem. GRASP: Gated regression-aware skill proposer for self-improving LLM agents. *arXiv preprint arXiv:2605.29668*, 2026.

Arjun Panickssery, Samuel R. Bowman, and Shi Feng. LLM evaluators recognize and favor their own generations. In *Advances in Neural Information Processing Systems*, volume 37, pp. 68772–68802, 2024. doi: 10.52202/079017-2197.

Valentina Pyatkin, Saumya Malik, Victoria Graf, Hamish Ivison, Shengyi Huang, Pradeep Dasigi, Nathan Lambert, and Hannaneh Hajishirzi. Generalizing verifiable instruction following. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://arxiv.org/abs/2507.02833.

Biswa Sengupta. Self-evolving agents with anytime-valid certificates. *arXiv preprint arXiv:2607.00871*, 2026.

Linfang Shang, Ming Xu, Yiding Sun, Tianle Xia, Lingxiang Hu, Lan Xu, and Ning Zheng. When self-evolution backfires: Pre-commit gating against skill contamination in LLM agents. *arXiv preprint arXiv:2608.05810*, 2026.

Shuai Shao, Qihan Ren, Chen Qian, Boyi Wei, Dadi Guo, Jingyi Yang, Xinhao Song, Linfeng Zhang, Weinan Zhang, Dongrui Liu, and Jing Shao. Your agent may misevolve: Emergent risks in self-evolving LLM agents. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2509.26354.

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

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 8634–8652, 2023. doi: 10.52202/075280-0377.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9440–9450, 2024a. doi: 10.18653/v1/2024.acl-long.511.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent Workflow Memory. *arXiv preprint arXiv:2409.07429*, 2024b.

Ian Waudby-Smith and Aaditya Ramdas. Estimating means of bounded random variables by betting. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 86(1):1–27, 2024. doi: 10.1093/jrsssb/qkad009.

Yifan Yang, Ziyang Gong, Weiquan Huang, Qihao Yang, Ziwei Zhou, Zisu Huang, Yan Li, Xuemei Gao, Qi Dai, Bei Liu, Kai Qiu, Yuqing Yang, Dongdong Chen, Xue Yang, and Chong Luo. SkillOpt: Executive strategy for self-evolving agent skills. *arXiv preprint arXiv:2605.23904*, 2026.

Xing Zhang, Yanwei Cui, Guanghui Wang, Ziyuan Li, Wei Qiu, Bing Zhu, and Peiyang He. The blind curator: How a biased judge silently disables skill retirement in self-evolving agents. *arXiv preprint arXiv:2607.07436*, 2026a.

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*, 2026b.

Xing Zhang, Yanwei Cui, Guanghui Wang, Ziyuan Li, Wei Qiu, Bing Zhu, and Peiyang He. Ratchet: How reliable must an LLM judge be to retire a skill? *arXiv preprint arXiv:2605.22148*, 2026c.

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*, 2026d.

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

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems*, volume 36, pp. 46595–46623, 2023. doi: 10.52202/075280-2020.

Terry Yue Zhuo, Minh Chien Vu, Jenny Chim, Han Hu, Wenhao Yu, Ratnadira Widyasari, Imam Nur Bani Yusuf, Haolan Zhan, Junda He, Indraneil Paul, Simon Brunner, Chen Gong, Thong Hoang, Armel Randy Zebaze, Xiaoheng Hong, Wen-Ding Li, Jean Kaddour, Ming Xu, Zhihan Zhang, Prateek Yadav, Naman Jain, Alex Gu, Zhoujun Cheng, Jiawei Liu, Qian Liu, Zijian Wang, Binyuan Hui, Niklas Muennighoff, David Lo, Daniel Fried, Xiaoning Du, Harm de Vries, and Leandro von Werra. BigCodeBench: Benchmarking code generation with diverse function calls and complex instructions. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=YrycTjllL0.

A Statistical Details

A.1 Reference identification and a randomized alternative

Let $\mathcal{F}_{i-1}$ contain the information available before outcome i, including its routed task and reference. Write $p_0'(x_i)$ for the reported no-skill success probability and c_i' for the conditional skill effect on that channel. The predictable-reference residual has mean

$$\mu_i' = \mathbb{E}[d_i' \mid \mathcal{F}_{i-1}] = c_i' + p_0'(x_i) - \widehat{p}_0(x_i).$$

Thus coverage of the judged effect requires control of reference error; interpreting true harm also requires the channel assumptions below. Appendix C.3 evaluates the reference empirically.

Randomization provides an alternative that removes the reference assumption. Let $A_i \in \{S, R\}$ be the skill (S) or reference/audit (R) assignment, with audit probability $q \in (0, 1)$. Write $y_i'^S, y_i'^R$ for the potential reported outcomes under each assignment. For a predictable outcome prediction $g_i \in [0, 1]$, define the augmented inverse-propensity weighted (AIPW) observation Z_i:

$$Z_i = \left(\frac{\mathbf{1}[A_i = S]}{1 - q} - \frac{\mathbf{1}[A_i = R]}{q} \right) (y_i' - g_i),$$
$$\mathbb{E}[Z_i \mid \mathcal{F}_{i-1}] = \mathbb{E}[y_i'^S - y_i'^R \mid \mathcal{F}_{i-1}],$$

where $\mathbf{1}[\cdot]$ is the indicator function. In the replay, g_i is the task's last reference estimate recorded before attempt i, or 0.5 if none exists. The confidence-sequence construction below applies after rescaling to range width $1/q + 1/(1-q)$. Small q gives audit outcomes large weights; Table 6 reports the power tradeoff for the tested estimator against shared and audit-only references.

A.2 Time-uniform target and interval construction

Let d_i' have conditional mean μ_i' and positive predictable weights λ_i. Define $\Lambda_t = \sum_{i \le t} \lambda_i$ after t observations. For skill j with error allocation δ_j, the interval covers the weighted conditional mean at every inspection time:

$$\bar{\mu}_t' = \frac{\sum_{i \le t} \lambda_i \mu_i'}{\Lambda_t}, \qquad \Pr(\exists t : \bar{\mu}_t' \notin [L_t, U_t]) \le \delta_j.$$

No stationarity is required. The target concerns the realized routed sequence, and uniformity permits both stopping and predictable selection of which skill receives the next observation.

Predictable plug-in empirical-Bernstein (PrPl-EB) interval. Write $p_i = \widehat{p}_0(x_i)$ and $\ell_j = \log(2/\delta_j)$. Since $d'_i \in [-p_i, 1-p_i]$, choose a center m_i in this unit-width interval and accumulate squared deviations v_i in V_i. Starting from $V_0 = 0$, set

$$m_i = \mathrm{clip}\left(\frac{\sum_{k<i} d'_k}{i},\, -p_i,\, 1-p_i\right), \qquad v_i = (d'_i - m_i)^2, \quad V_i = V_{i-1} + v_i,$$

$$\lambda_i = \min\left\{0.75,\, \sqrt{\frac{2\ell_j}{(1/4 + V_{i-1})\log(1+i)}}\right\}.$$

Here clip truncates to the stated endpoints; dividing by i includes one initial zero residual for stabilization. With penalty $\psi_E(\lambda) = -\log(1-\lambda) - \lambda$, compute the weighted center C_t and half-width W_t:

$$C_t = \frac{\sum_{i\le t} \lambda_i d'_i}{\Lambda_t}, \qquad W_t = \frac{\ell_j + \sum_{i\le t} v_i \psi_E(\lambda_i)}{\Lambda_t},$$

$$[L_t, U_t] = [C_t - W_t,\, C_t + W_t] \cap [-1, 1].$$

We use the predictable plug-in construction of Waudby-Smith & Ramdas (2024), based on empirical-Bernstein supermartingales (Howard et al., 2021). The weight expression substitutes the variance estimate $\widehat{\sigma}^2_{i-1} = (1/4 + V_{i-1})/i$ into the usual schedule. Empty sequences use $[-1, 1]$; the unweighted sample mean reported as a diagnostic need not equal C_t.

For total error budget $\delta = 0.05$ and a cap of $J_{\max} = 200$ skills, each skill receives $\delta_j = \delta/J_{\max}$. The family-wise statement follows by a union bound over these tested skills, including injected plants. Synthesis stops accepting new skills at that cap.

A.3 Judge channel and threshold guarantees

For arm $A \in \{S, R\}$, let ρ^A_{FP} and ρ^A_{FF} be its false-pass and false-fail probabilities. Define $\kappa_A = 1 - \rho^A_{\mathrm{FP}} - \rho^A_{\mathrm{FF}}$ and $\Delta_0 = \rho^S_{\mathrm{FP}} - \rho^R_{\mathrm{FP}}$. Here y^A denotes true correctness and y'^A its reported label. The residuals d, d' use matched population references, so their means equal the true and judged arm differences. The finite-reference qualification is given in Appendix A.1.

Lemma 1 (Affine judge channel)**.**

$$\mathbb{E}[d'] = \kappa_R \mathbb{E}[d] + (\kappa_S - \kappa_R)\mathbb{E}[y^S] + \Delta_0.$$

Proof. For either arm, $\mathbb{E}[y'^A] = \kappa_A \mathbb{E}[y^A] + \rho^A_{\mathrm{FP}}$. Subtract the reference-arm mean and add and subtract $\kappa_R \mathbb{E}[y^S]$. □

Proposition 1 (Fixed-point retirement)**.** *Assume $\mathbb{E}[d'_i \mid \mathcal{F}_{i-1}] = \kappa \mathbb{E}[d_i \mid \mathcal{F}_{i-1}] + \Delta_{\mathrm{eff}}$ for every i, with constant $\kappa > 0$ and $\Delta_{\mathrm{eff}} \ge 0$. Let $\bar{\mu}_t$ denote the weighted true mean, using the same weights as the judged mean $\bar{\mu}'_t$. If an anytime-valid confidence sequence reports $U_t < 0$, then, with probability at least $1-\delta$, the same weighted true mean is below $-\Delta_{\mathrm{eff}}/\kappa \le 0$.*

Proof. On the simultaneous coverage event, $\bar{\mu}'_t \le U_t < 0$. Affinity commutes with weighting: $\bar{\mu}'_t = \kappa \bar{\mu}_t + \Delta_{\mathrm{eff}}$. Rearranging gives the claim, including at the first crossing. □

Under unequal attenuation, the conditional effective offset satisfies

$$\Delta_{\mathrm{eff},i} = (\kappa_S - \kappa_R)\mathbb{E}[y^S_i \mid \mathcal{F}_{i-1}] + \Delta_0,$$

$$\Delta_0 + \min(0, \kappa_S - \kappa_R) \le \Delta_{\mathrm{eff},i} \le \Delta_0 + \max(0, \kappa_S - \kappa_R).$$

The same argument applies when $\kappa_R > 0$ is constant and every $\Delta_{\mathrm{eff},i} \ge 0$, using their weighted average.

Proposition 2 (Negative-bar admission is not invariant)**.** *Under an arm-blind channel with constant $0 < \kappa \le 1$, the threshold $L_t \ge b$ certifies $\bar{\mu}_t \ge \max(-1, b/\kappa)$. The untruncated threshold b/κ equals b for every κ exactly when $b = 0$. At $b = -\tau_{\mathrm{adm}}$, the bound becomes vacuous once $\kappa \le \tau_{\mathrm{adm}}$.*

Proof. Coverage and $L_t \geq b$ imply $\kappa\bar{\mu}_t = \bar{\mu}'_t \geq b$. Divide by $\kappa > 0$ and use $\bar{\mu}_t \in [-1, 1]$. Zero preserves non-negativity; a positive b strengthens the true lower bound as attenuation increases. □

Why homogeneous attenuation matters. Arm blindness within each task does not preserve the sign after averaging when κ varies by task. Equal-frequency effects $-0.2, +0.3$ average to $+0.05$, but attenuations $0.9, 0.1$ produce a judged mean of -0.075. The propositions require the stated constant channel relation for the same weighted target.

B Experimental Protocol

B.1 Lifecycle configuration and schedule

The audit probability is $q = 0.15$. The family-wise error budget is $\delta = 0.05$ over $J_{\max} = 200$ skills. Curated libraries hold at most 50 active skills; probation uses ten shadow slots by default. Append-only is uncapped. The expanded natural-skill study uses 40 shadow slots (Appendix C.5). Probation prioritizes shadow skills with the fewest attempts in the round toward a quota of three each, subject to applicable tasks arriving. The default budget is 60 skill-arm attempts; extended-budget and utility/racing studies use 180. The default tolerance $\tau_{\text{adm}} = 0.15$ was fixed after pilot power checks. Positive-utility studies use a small chosen margin $b = +0.02$; Appendix C.4 reports tolerance sensitivity.

At capacity, the count arm evicts the lowest-scoring skill, prioritizing those eligible for retirement and breaking score ties by least recent use. PAWL evicts the active skill with the smallest lower bound. Capacity evictions are distinct from certified retirements.

In Algorithm 1, the curator reads accumulated evidence after each round's training and synthesis. The attempt budget limits skill solves, including any without a reference; it does not guarantee a decision or expire an unrouted candidate. Stored-sequence replays inspect every attempt prefix; their first crossings precede the actual retirements by a median of two attempts.

Most runs last 60–200 rounds; the long-horizon study lasts 1,000 (Appendix C.6).

B.2 Feedback and reference initialization

Each task's reference is the mean of its retained no-skill outcomes, initialized with clean executable seed labels. Cold start drops those seeds once five in-run no-skill draws exist; the mix policy keeps them. Earlier residuals remain in the cumulative sequence. For a task with true no-skill success probability p_0 and clean-seed fraction w, false-pass corruption at rate r adds offset $wr(1 - p_0)$ relative to a fully matched judged reference; false-fail corruption adds $-wrp_0$. These initial labels supplement the executable feedback paths described in Section 5. Across the 150-round PAWL planted-harm studies, unanswered judgments account for 0.30% of Opus calls and less than 0.01% of GLM-5 calls.

B.3 Baselines, data splits and evaluation

The success-count baseline (Zhang et al., 2026c) assigns $+1$ to a reported pass and -1 to a reported failure. With S_s, F_s their counts over $n_s = S_s + F_s$ uses, N_s suite tasks since creation, and $\widehat{p}_s = S_s/n_s$ the reported success rate, its statistic is

$$\frac{S_s - F_s}{N_s} = a_s(2\widehat{p}_s - 1).$$

This instantiates Equation (1) with $z_{s,i} = 2y'_i - 1$. Retirement requires $N_s \geq 100$ and a score at most -0.10. The count baseline can restore earlier libraries using clean development scores; PAWL does not.

The point-estimate admission ablation uses the unweighted residual mean $\bar{d}'_n \geq -0.15$ after $n \geq 20$ observations, retaining the same paired measurements and confidence-bound retirement rule.

The SkillOpt/GRASP-style gate is adapted to the shared lifecycle and tests skills on every reserved validation (oracle) task, regardless of applicability. It thresholds the mean judged skill-minus-no-skill

effect: admit at ≥ -0.15, reject below it, and retire active skills below -0.10. Each reading uses one solve per tested skill and one shared no-skill solve per task. Retirement studies read every five rounds; admission studies read every round and test at most eight skills, prioritizing the most-used active or shadow skills. With ten MBPP+ or five IFBench oracle tasks, effects lie on grids of 0.1 or 0.2; the two cutoffs have no intervening grid value. The gate never reads the test split.

The MBPP+ slice selects 150 tasks from no-skill failures across three solver families, prioritizing primary-model failures. A fixed stratified split allocates 80 train, 20 dev, 40 test and ten oracle tasks. IFBench retirement uses tasks with at least 60% no-skill success over five probes, with 60 training, 20 development and five oracle tasks. The IFBench and BIRD utility studies use failure-selected 100-task pools, with fixed splits of 60 training, 20 development, 15 test and five oracle tasks.

Naming and instruction metadata. Every MBPP+ condition supplies the required entry-point name. Otherwise, identifier rules can manufacture a utility gain: on a 40-task failure-selected slice, revealing the name raises no-skill success from 0.23 to 0.62. IFBench supplies instruction-family names and routing tags, so its results use metadata-assisted routing. Unseen instruction types in the held-out split are absent from training, not hidden at evaluation.

Held-out value and audit units. Frozen libraries and no-skill references are evaluated with $k = 3$ solves per task per arm on identical held-out tasks. The utility comparisons in Table 9 use 228 tasks for MBPP+, 200 each for IFBench and BIRD, and 163 for HumanEval+. MBPP+ and HumanEval+ evaluations remove planted skills from the frozen libraries. Each 200-task set is disjoint from all four splits of its 100-task development pool. Utility intervals use 10,000 bootstrap resamples of tasks with all their seeds; Figure 3C uses task-seed resampling for the separate racing studies. Test outcomes never enter synthesis, routing, curation or checkpoint selection.

Wilson intervals describe pooled plant outcomes, without correcting for dependence within runs or repeated canary designs. Retirement audits average distinct routed tasks equally against pooled observed no-skill outcomes; those weights and references differ from the certificate's predictable attempt-level target. These audits are descriptive checks rather than simultaneous causal certificates at adaptive decision times.

B.4 Canary interventions and screening

Designed canaries are hand-written instructions presented as reusable guidance. Harmful examples assert a stale grader convention that conflicts with the actual task; controls give ordinary programming advice. The primary MBPP+ set changes the function name, packs arguments into one tuple, or moves the function into a class, with screened effects -0.531, -0.312 and -0.688. IFBench combines lowercasing, answer fencing and a follow-up question, with effects -0.38, -0.34 and -0.42. These instructions are inserted at rounds 10, 25 and 40, active in retirement studies and in probation in admission studies. Repeated seeds reuse the same designs; Table 3 lists representative instructions and their screens.

Named versus omitted harm. The judge receives the required function name, so a renamed entry point conflicts with an explicit prompt field. The omitted-harm set instead changes return type, wrapping or precision while preserving the name and parameters. Those requirements have no dedicated field in the testless prompt and must be inferred from the task description. The contrast concerns visibility to the judge, not whether the task implies the correct behavior.

Designed-canary selection. These screens use Opus 4.7 and executable grading, comparing skill and no-skill solve rates on tasks with positive preliminary no-skill success. The primary MBPP+ harmful screen forces each candidate onto 16 tasks at $k = 2$; IFBench uses 20 tasks at $k = 5$. Both retain candidates with mean effect at most -0.30. The omitted-harm screen first applies the router's applicability gate to a 40-task pool and uses $k = 8$ on accepted tasks: string conversion, list wrapping and rounding cover 27, 30 and eight tasks, respectively. Their effects are -0.292, -0.488 and -0.109; the two weaker candidates are retained deliberately to span severity.

The neutral screen forces candidates onto 20 tasks at $k = 8$ and retains estimated effects $|\widehat{c}| \leq 0.10$. It also requires correlation at least 0.70 with the common per-task effect profile, the average per-task effect across the 14 screened candidates, and residual standard deviation at most 0.296 after

Table 3: **Representative planted instructions and their screens.** Instructions are verbatim excerpts; applicability is summarized. m: screened tasks; k: solves per task per arm; SE: standard error across paired task effects.

Family and instruction excerpt	Applicability	Effect $\pm$ SE	$m \times k$
MBPP+, named harm. "Confirm the only top-level function is named `solution`."	Tasks asking for a function	-0.531 ± 0.148	16×2
MBPP+, omitted harm. "Return `[answer]` on the last line. Keep the function name and the parameters exactly as the task states."	Functions returning a value	-0.488 ± 0.078	30×8
MBPP+, weaker omitted harm. "Apply it to numbers inside a returned collection as well." (The instruction rounds results to two decimal places.)	Numeric answers, including collections	-0.109 ± 0.130	8×8
IFBench, formatting harm. "Write the whole answer in lowercase, including sentence openings and proper nouns, and make the last line a single short question that checks the answer met the need."	Every request	-0.420 ± 0.126	20×5
MBPP+, screened neutral. "Deal with each special case at the top of the function and return from it at once, so the rest of the body is the ordinary path at one level of indentation."	Tasks with special cases such as empty input	-0.025 ± 0.081	20×8

subtracting that profile, to exclude unusually content-dependent effects. These labels describe the screening distribution, which can differ from the tasks later selected by the router.

Natural-skill controls. Candidate selection starts from skills synthesized in earlier clean-feedback runs with at least 30 observations and a lower bound above $+0.10$ or upper bound below -0.10, supplemented by skills from false-pass runs with negative executable paired effects. A fresh executable screen evaluates the original instructions with and without the skill on the tasks reached in its source run: $k = 16$ solves per task per arm for at most two tasks, and $k = 8$ otherwise. Labels are fixed before insertion: helpful at estimated effect $\geq +0.15$, harmful at ≤ -0.15; intermediate candidates are excluded. The selected skills cover one to thirteen source tasks each. MBPP+ includes six skills per class, with screened effects $+0.22$ to $+0.94$ and -0.69 to -0.16; the helpful designs share seven source tasks. IFBench combines six helpful synthesized skills with the three designed harms above. Helpful and harmful plants alternate from round 10 under the two schedules in Appendix C.5.

B.5 Generation settings and prompts

Model weights remain fixed. BIRD-SQLite uses Kimi K2.5; the other primary studies use Opus 4.7, with additional solvers in Table 2. Opus 4.7 leaves temperature and top-p unset and uses an 8,192-token output cap; Kimi K2.5 and GLM-5 use model-default sampling with thinking disabled; GPT-5.5 uses low reasoning effort. Judges retain their model's sampling settings; Sonnet 4.6 offline judging uses temperature zero.

Skill synthesis (condensed prompt). Synthesis proposes at most two skills per round from failure clusters. The user message supplies representative task inputs, attempted outputs, executable errors, existing skills, authoring guidance and any retry feedback. Its core instructions, condensed, are: "Write one guidance skill from the failure cluster. Its text will guide a future solver on similar tasks. Describe a reusable task class, without task-identifier lookups. Give two to four abstract routing cues, applicability conditions, and a key insight or pitfalls. Follow the authoring guidance and avoid duplicating existing skills. Return a structured skill within 1,500 characters." The requested guidance also includes checks before returning an answer.

Judge prompts. The testless judge sees the task, required function name and candidate solution without unit tests; the assert variant also shows one example assertion. Judge inputs retain up to

2,000 task, 4,000 solution and 600 assertion characters. System messages below are verbatim; user placeholders and spacing are condensed.

Testless: system message

You grade candidate Python solutions to programming tasks. You are given the task description and one candidate solution. Decide whether the solution is correct: whether it computes what the task asks for, for every valid input, not merely whether it looks reasonable. You do not have the unit tests. Answer with exactly one word, PASS or FAIL, on the first line, then at most one short sentence of reason.

Assert: system message

You grade candidate Python solutions to programming tasks. You are given the task description, one example assertion the solution must satisfy, and one candidate solution. Decide whether the solution is correct for every valid input, not only for the example. You do not have the full test suite. Answer with exactly one word, PASS or FAIL, on the first line, then at most one short sentence of reason.

Shared user message

Task: *[task description]*

Required function: *[function name]*

Example assertion the solution must satisfy: *[example assertion]* *Assert only*

Candidate solution: *[candidate solution in a Python fenced code block]*

Is this solution correct? Answer PASS or FAIL.

C Supporting Evidence

C.1 Cohort robustness

Eligibility in Table 4 is independent of retirement outcome, using the matching procedure in Section 5.

Table 4: **Retirement sensitivity to follow-up length.** Retired/all harmful plants on MBPP+; the 20-round requirement applies to both arms.

	All plants		At least 20 rounds	
Channel	**Count**	**PAWL**	**Count**	**PAWL**
Clean	6/9	7/9	5/6	6/6
Noise 0.30	3/9	3/9	2/6	2/6
False pass 0.45	2/18	16/18	0/12	12/12
False pass 0.70	0/18	8/18	0/12	8/12
False fail 0.30	6/9	7/9	5/6	5/6

For the two false-pass channels, the paired bootstrap enumerates all ordered resamples of the six seed clusters with replacement, keeping every plant and both arms within each sampled cluster. The interval takes the 2.5th and 97.5th percentiles of the pooled difference in retirement fractions. It describes between-seed variability conditional on the repeated canary designs, not generalization to unseen harms.

Screen-labelled non-harmful controls. The primary MBPP+ controls comprise nine neutral and three helpful designs across three seeds. The false-fail channel leads success counts to retire 7 of 36 matched controls, against 2 of 36 for PAWL. On clean feedback, the outcomes are 0 of 36 and 4 of 36, respectively. These labels come from the pre-insertion screen, whose task distribution can differ from later routing (Appendix B.4).

C.2 Baseline and confidence-bound comparisons

Stored-sequence comparisons pool 47 MBPP+ and IFBench runs, comprising 1,639 skill sequences and 134,369 eligible observations. The synthesized-skill cohorts contain 1,092 and 356 sequences, respectively; the remaining sequences come from 98 harmful and 93 screen-labelled non-harmful

canaries. Table 1 isolates denominator repair and fixed-budget testing. The 31 harmful false-pass 0.70 sequences pool MBPP+ primary, longer-horizon and single-plant exposure studies, including both reference initialization policies.

Reported labels are reconstructed as $y'_i = d'_i + \widehat{p}_0(x_i)$. With $\bar{y}'_n$ the mean reported outcome over n observations, the use-normalized rule checks $2\bar{y}'_n - 1 < -0.10$. The fixed-budget paired rule retires when $U_n^{\text{fixed}} < 0$, using the same per-skill error allocation and a single-look Hoeffding upper bound:

$$U_n^{\text{fixed}} = \bar{d}'_n + \sqrt{\frac{\log(2/\delta_j)}{2n}}.$$

Each residual has predictable range width one. The fixed test spends its error budget at one reading; PAWL covers every inspection. Budgets 20, 60 and 180 are retrospective replay settings. All rules receive the same recorded observations, without additional solves. Early PAWL retirements can shorten sequences and remove them from later-budget comparisons; the lower block therefore describes rule behavior conditional on available evidence, not alternative full-loop detection rates. At budgets 20, 60 and 180, the excluded early PAWL retirements comprise 2, 20 and 41 harmful sequences and 0, 0 and 5 screen-labelled non-harmful sequences, respectively.

Confidence bounds. The comparisons in Table 5 use the same per-skill error allocation unless otherwise marked.

Table 5: **Confidence bounds in replay (A) and the full loop (B).** Parentheses give median harmful-detection attempts. Admissions use $L \geq -0.15$ on 1,448 synthesized skills. Follow-up eligibility in B applies to both arms.

A. Stored-sequence replay **Rule**	**Harmful retired**	**Non-harmful retired**	**Admitted**
PrPl-EB ($\delta/200$)	66/98 (116)	6/93	152
Stitched Hoeffding ($\delta/200$)	75/98 (75)	6/93	199
Fixed-n normal, repeatedly read after $n \geq 20$	81/98	15/93	697
No union correction (δ per skill)	80/98	11/93	586

B. End-to-end comparison **Outcome (rounds completed)**	**Bound**	**All plants**	**At least 20 rounds**
Clean, harmful (60)	PrPl-EB	4/6	4/4
Clean, harmful (60)	Hoeffding	6/6	4/4
False pass 0.70, harmful (60)	PrPl-EB	4/6	4/4
False pass 0.70, harmful (60)	Hoeffding	3/6	3/4
Clean, non-harmful (80)	PrPl-EB	2/18	2/18
Clean, non-harmful (80)	Hoeffding	2/18	2/18

The normal-approximation interval for the sample mean in Table 5A is not anytime-valid under repeated inspection. Allocating the full δ to each skill also removes the family-wise guarantee. Panel B changes the boundary inside the lifecycle, allowing retirement timing to change future routing. It tests a different question from replaying a fixed sequence.

The adapted gate (Appendix B.3) retires all nine harmful plants in both the clean and false-pass 0.45 studies, but also two of nine screened non-harmful controls under false pass.

C.3 REFERENCE ESTIMATION AND DECISION AUDITS

Fresh skill/no-skill evaluations on recorded routed tasks use task-use weights and normal-approximation 95% intervals conditional on those tasks. The in-loop estimate agrees in sign for 17/17 effects whose fresh intervals exclude zero. All 29 fresh estimates lie within the in-loop intervals, with Pearson correlation 0.975. None of these skills has an in-loop upper bound below zero, so this checks agreement away from retirement crossings.

Table 6: **Reference-estimator replays at two audit rates.** Entries are retirements/plants; controls are screen-labelled non-harmful. Audit-only references use prior randomized audits. The $q = 0.5$ cohort contains six MBPP+ runs with clean or false-pass 0.45 feedback.

	Audit rate $q = 0.15$		**Audit rate** $q = 0.5$	
Estimator	**Harmful**	**Non-harmful**	**Harmful**	**Non-harmful**
PAWL (shared reference)	$66/98$	$6/93$	$15/18$	$0/7$
Audit-only reference	$24/98$	$4/93$	$6/18$	$0/7$
Randomized AIPW	$1/98$	$0/93$	$1/18$	$0/7$

Executable audits of retirement decisions. The retirement audit covers a separate set of 29 synthesized skills from the MBPP+ channel studies, with a median of three routed tasks per skill (range one to six). Their task-averaged executable contribution has median -0.36. Four task-bootstrap intervals lie below zero and 17 cover it; eight single-task cases have no interval. The two unresolved Opus planted-harm audits in Section 6.2 are separate: one interval covers zero and the other is unavailable. These descriptive audits use the units specified in Appendix B.3.

C.4 ADMISSION AND TOLERANCE

The designed-harm study tests resistance to below-bar admission (Table 7); the natural-skill study below also tests useful acceptance.

Table 7: **Admission outcomes on designed harmful candidates.** All 42 candidates per rule are included. Below-bar admissions have screened effects below -0.15; within-bar admissions remain negative.

Rule	**Below-bar admit**	**Within-bar admit**	**Expired**	**Certified retire**	**Unresolved**	**Median n at decision**
PAWL	0	0	13	15	14	63
Point	8	2	3	27	2	51

In the adapted gate's separate admission test at false-pass 0.70, six of nine candidates are admitted below the bar; all six admissions are later reversed, with a median delay of 2.5 rounds. A separate replay varies the bar on fixed observations: no harmful sequence reaches $L \geq -\tau_{\mathrm{adm}}$ for $\tau_{\mathrm{adm}} \leq 0.15$, compared with 7 of 98 at $\tau_{\mathrm{adm}} = 0.25$ and 38 at $\tau_{\mathrm{adm}} = 0.50$.

C.5 NATURAL-SKILL CONTROLS

Table 8 extends Figure 2C to both suites and channels.

The screened populations in Appendix B.4 enter probation under each admission rule with clean or false-pass 0.70 feedback, across three seeds. The default cohort uses ten shadow slots, insertions every five rounds from round 10, and 100 rounds; the expanded cohort uses 40 slots, insertions every round and 150 rounds. Rules share each cohort's schedule and population; capacity, timing and duration change together between cohorts.

On biased MBPP+, PAWL admits $3/3$ plants each for wind-chill, rounding and recurrence, and $2/3$ each for regex search, return structure and substring matching. The stricter-screen analysis in Section 6.2 excludes the return-structure and substring designs.

The 15 helpful MBPP+ admissions under bias accumulate 493 residuals through their admission rounds. Matching recorded reference sizes to clean-seed counts and prior in-run updates shows that all 493 exclude clean seeds; their counts agree with the saved decision sequences. Thus clean seeds do not directly enter these admission residuals; finite-reference uncertainty remains.

Table 8: **Admission and subsequent outcomes with expanded probation.** Counts pool three seeds at 150 rounds. Later retirements are a subset of ever-admitted skills; other outcome columns are disjoint.

Suite, channel	Rule	Ever admitted	Later retired	Rejected	Expired	Open	Merged
A. Helpful synthesized skills: 18 per row							
MBPP+, clean	PAWL	18	0	0	0	0	0
MBPP+, clean	Point	18	0	0	0	0	0
MBPP+, clean	Gate	1	0	0	8	9	0
MBPP+, FP 0.70	PAWL	15	0	0	0	3	0
MBPP+, FP 0.70	Point	18	0	0	0	0	0
MBPP+, FP 0.70	Gate	17	16	1	0	0	0
IFBench, clean	PAWL	8	0	0	4	6	0
IFBench, clean	Point	14	0	0	0	3	1
IFBench, clean	Gate	14	14	4	0	0	0
IFBench, FP 0.70	PAWL	3	0	0	12	3	0
IFBench, FP 0.70	Point	14	0	0	0	3	1
IFBench, FP 0.70	Gate	18	18	0	0	0	0
B. Harmful skills: 18 per MBPP+ row; 9 per IFBench row							
MBPP+, clean	PAWL	0	0	13	3	2	0
MBPP+, clean	Point	4	1	13	0	1	0
MBPP+, clean	Gate	0	0	0	11	7	0
MBPP+, FP 0.70	PAWL	0	0	0	17	1	0
MBPP+, FP 0.70	Point	13	7	0	5	0	0
MBPP+, FP 0.70	Gate	17	17	1	0	0	0
IFBench, clean	PAWL	0	0	0	0	9	0
IFBench, clean	Point	0	0	1	0	8	0
IFBench, clean	Gate	0	0	9	0	0	0
IFBench, FP 0.70	PAWL	0	0	0	1	8	0
IFBench, FP 0.70	Point	4	2	0	1	3	1
IFBench, FP 0.70	Gate	3	3	6	0	0	0

Open denotes unresolved probation; merged denotes duplicate removal before admission. Rejection is a confidence-bound or gate decision, whereas expiry exhausts the evidence budget. Harmful MBPP+ skills are synthesized; harmful IFBench skills are designed formatting interventions. Gate denotes the adapted SkillOpt/GRASP-style held-out rule.

The gate evaluates $1/18$ helpful MBPP+ plants under clean feedback and $18/18$ under bias. The 17 unread clean-feedback plants account for all eight expirations and nine unresolved outcomes. Clean runs synthesize 112–140 skills, compared with 33–76 under bias, competing for the same eight gate readings per round.

On clean IFBench, every harmful plant remains unresolved under PAWL and eight of nine do under the point rule, with at most 25 attempts each. The default cohort adds a queueing cost: under PAWL, 47 of 72 MBPP+ plants across both classes and channels never enter probation within 100 rounds.

C.6 JUDGE DISTORTION AND HORIZON SENSITIVITY

In the live planted-harm studies, the judged-minus-executable residual is -0.14 to -0.18 for named harm and -0.10 to -0.16 for omitted harm across four study cells; arm-blindness predicts $+0.24$ to $+0.47$. The negative differences indicate a relative penalty on skill-arm outputs.

Figure 3A applies equal-arm (κ, Δ_0) channels to 500 stored sequences, with five relabellings per channel. The audit proxy includes the 343 sequences whose executable intervals do not lie below zero. Its first retirement occurs at $\Delta_0 = -0.05$; rates exceed 5% near -0.20 to -0.30. This 5% cutoff describes replay frequencies and is distinct from the family-wise error budget.

Offline relabelling by Opus 4.7, GLM-5 and Sonnet 4.6 gives false-pass offsets $\Delta_0 = -0.022$, -0.041 and -0.048 on synthesized skills. All three yield zero audit-proxy retirements in their own replay cells.

Observed long-horizon decisions. Two 1,000-round MBPP+ runs use the live Opus 4.7 judge and an uncapped exposure floor targeting three attempts per unresolved skill per round. Each inserts the same nine screened near-neutral designs, active in groups of three at rounds 10, 25 and 40 (Appendix B.4). Retirements rise from $4/18$ by round 150 to $12/18$ by round 1,000; six plants remain active after 1,744–10,993 observations. The screen labels and task-averaged executable audits concern different targets from the certificate (Appendix B.3), so these counts do not estimate a false-retirement rate.

C.7 Evidence allocation and held-out utility

Exposure and racing. In the three displaced-skill IFBench runs, the unresolved skill receives no attempts during the final ten rounds, with only 46, nine and two observations accumulated after 200 rounds. The exposure-floor variant keeps sampling and retires all three at a median of 372 attempts. A second three-seed study retires the first plant at rounds 37–130; of the six later plants, five are merged as duplicates and one remains active.

The separate 100-round MBPP+ exposure-band study uses the mixed reference and targets three to eight attempts per unresolved active skill per round. Both arms retire all nine harmful plants, but median detection increases from 5 to 26 rounds (46 to 103 attempts). Its post-planting drop is the preceding five-round mean development score minus the minimum over the next 20 rounds. Despite shallower drops, the final ten-round mean development score is 0.620 with the band and 0.630 without it.

All 220 racing cuts across the two thresholds are heuristic, with no certified eliminations. At the positive-utility bar, racing reduces budget expiries from 27 to two. Candidates are ranked on their own routed tasks, so this allocation also reflects task difficulty and available improvement, not a common-task ranking of skill quality.

Held-out utility and task reach. Held-out effects are in Table 9. MBPP+ and HumanEval+ have little headroom, with no-skill success of 0.997 and 0.957, respectively.

Table 9: **Held-out utility and library size at the longest matched horizon.** Effects pool three seeds; intervals use task-clustered bootstraps. Active counts are per-seed ranges; fixed/broken counts show corrections/regressions versus no skills across task–seed pairs.

Suite	Lifecycle	Active	Paired effect [95% CI]	Fixed	Broken
MBPP+	Append-only	30–94	-0.0034 $[-0.0093, +0.0029]$	6	9
MBPP+	Success count	47–48	-0.0024 $[-0.0083, +0.0024]$	4	7
MBPP+	PAWL	4–6	$+0.0005$ $[-0.0034, +0.0049]$	5	3
IFBench	Append-only	61–70	$+0.0161$ $[-0.0083, +0.0417]$	69	52
IFBench	Success count	47–50	-0.0022 $[-0.0289, +0.0250]$	69	72
IFBench	PAWL	2–6	$+0.0044$ $[-0.0211, +0.0300]$	65	58
BIRD-SQLite	Append-only	29–38	$\mathbf{-0.0261}$ $[\mathbf{-0.0517}, \mathbf{-0.0006}]$	68	100
BIRD-SQLite	Success count	21–40	-0.0006 $[-0.0244, +0.0256]$	74	76
BIRD-SQLite	PAWL	1–2	$+0.0011$ $[-0.0206, +0.0239]$	70	68
HumanEval+	Append-only	30–94	$+0.0034$ $[-0.0150, +0.0218]$	13	9
HumanEval+	PAWL	4–6	$+0.0116$ $[-0.0020, +0.0293]$	14	5

Across the tested MBPP-trained libraries, pooled router engagement is 55.7% on held-out MBPP+ and 0.2% on BIRD-SQLite.