LOOKAHEAD-GUIDED RECURSIVE SELF-IMPROVEMENT FOR LANGUAGE MODEL AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recursive self-improvement can reduce the manual effort needed to build capable language model agents, but edits that help the current agent can become harmful after later modifications. We aim to select currently beneficial edits that retain their value as the agent continues to change. We introduce Lookahead-Guided Recursive Self-Improvement (LRSI), which measures each candidate's contribution in sampled future agent versions by enabling and disabling it while holding subsequent modifications fixed. LRSI selects the largest immediate improvement among candidates whose average contribution in the least favorable sampled continuations is nonnegative, keeping the language model fixed and committing only the current edit. Across ALFWorld, Terminal-Bench 2.0, SWE-bench Verified, and Aider Polyglot with three model backbones and matched optimization budgets, LRSI improves task success, pass, or resolved rates by 2.39–4.39 percentage points over the strongest evaluated baseline in each setting. By distinguishing an edit's continued usefulness from the overall quality of future agents, this work provides a testable criterion for selecting improvements that better withstand subsequent self-modification.

1 INTRODUCTION

Large language model (LLM) agents combine reasoning with tool use to carry out multi-step tasks, extending language models from answering questions to acting in external environments (Yao et al., 2023). Their reliability in applications such as software development depends on the *harness*: the prompts, tools, and control logic surrounding the model. Automating improvements to this harness can reduce manual engineering without retraining the model, but repeated self-modification raises a challenge: changes that help one version may interact adversely with later changes.

GEPA refines prompts through reflection on execution traces (Agrawal et al., 2026), while Meta-Harness uses accumulated experience to search over harness code (Lee et al., 2026). Self-Harness further subjects proposed changes to regression tests before accepting them (Zhang et al., 2026a). These mechanisms evaluate improvements to the current agent, leaving their persistence under later modifications open. For example, a retry rule that initially recovers failed tool calls may waste the execution budget after a later controller introduces its own retries (Figure 1). Reliable self-improvement therefore requires evaluating how an edit's usefulness changes as the agent evolves.

Search over future agent versions already addresses part of this challenge. DGM preserves alternative lineages and stepping stones beyond the current best agent (Zhang et al., 2026b); HGM uses descendant outcomes to identify agents with greater self-improvement potential (Wang et al., 2026b). These approaches guide the search toward promising descendants; the contribution of each earlier change within a descendant is a separate quantity. EvolveNet also addresses interference when composing independently developed adaptations (Nie et al., 2026), but validating an available composition leaves open what happens under subsequent, unseen modifications. Indeed, overall progress can conceal deterioration in an earlier edit's usefulness because new functionality may compensate for the harm it introduces. What remains unresolved is how to account for this changing usefulness when deciding which currently beneficial edit to accept.

We study whether anticipating an edit's contribution after subsequent self-modification can improve recursive harness optimization under a fixed resource budget. Our central distinction is between an agent becoming better and a particular change continuing to help that agent. We introduce

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.

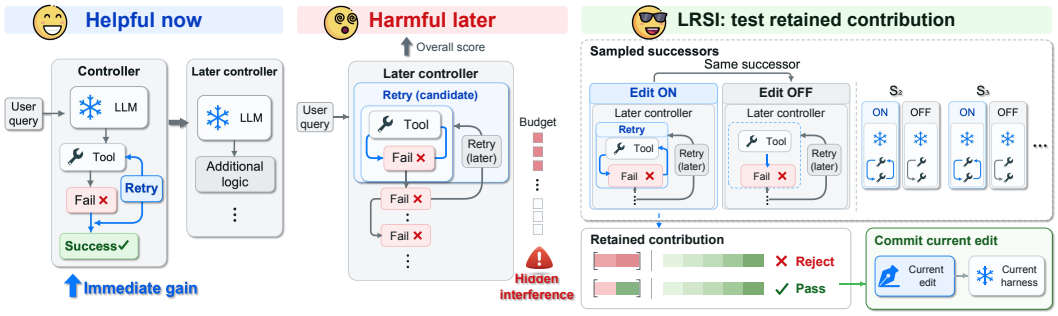


Figure 1: **An edit that helps now may become harmful later.** A useful retry rule can conflict with a later controller, even when the overall agent improves. LRSI compares the same sampled successors with the candidate edit on and off, keeps candidates whose average contribution in the least favorable sampled continuations is nonnegative, and commits the one with the largest immediate gain.

Lookahead-Guided Recursive Self-Improvement (LRSI), which evaluates currently useful candidates inside sampled future versions, comparing each version with and without the candidate while holding the subsequent modification fixed. It selects the largest immediate improvement among candidates whose least favorable sampled continuations retain nonnegative mean contribution. The base model and proposal generator remain fixed, and only the selected current edit is committed; future modifications serve as probes for the present decision.

Our contributions are threefold:

- **We reveal a gap between immediate improvement and retained usefulness.** In a retrospective ALFWorld audit, 18.2% of 336 edit–successor pairs have negative retained contributions despite positive immediate gains (Section 3). Further analyses show that both improving successors and positive average contributions can conceal harmful interactions.
- **We propose a contribution-aware criterion for recursive self-improvement.** LRSI makes resilience to adverse subsequent modifications part of current edit selection. We characterize the robustness of its population criterion to shifts in the successor distribution, with explicit conditions for nonnegative expected retained contribution (Section 4.4).
- **LRSI improves performance across tasks and backbones.** In the full-harness comparisons, LRSI exceeds the strongest evaluated baseline in each setting by 2.39–4.39 percentage points across four benchmarks and three backbones under matched optimization budgets (Tables 1 and 2). An independent ALFWorld audit also finds fewer observed contribution reversals under fresh continuations: 6.0% for LRSI versus 19.9% for current-gain selection, among valid contrasts in their selected cohorts (Section 5.3).

2 Related Work

From reflection to harness optimization. Reflective optimization converts execution feedback into improved prompts through evolutionary search in GEPA and structured reflection in MARS (Agrawal et al., 2026; Hou et al., 2026). Executable search broadens the editable surface: SICA modifies its implementation, while Meta-Harness and VeRO support trace-guided optimization of complete agents (Robeyns et al., 2025; Lee et al., 2026; Ursekar et al., 2026). These approaches expand how improvements are generated and evaluate them through whole-agent scores. Measuring an edit's continued usefulness requires comparison within the same successor. LRSI adds this selection criterion, keeping the base model and proposal generator fixed.

Searching for future improvement. DGM maintains an archive to exploit stepping stones beyond the current best agent (Zhang et al., 2026b). HGM estimates clade metaproductivity from descendant outcomes (Wang et al., 2026b), while MetaSkill-Evolve jointly evolves task skills and their improvement policies (Wang et al., 2026c). Descendant quality and improvement-policy productivity assess an agent's or search policy's promise. LRSI instead toggles only the candidate within

fixed sampled successors, separating its retained marginal contribution from total successor utility to guide current acceptance.

Regression control and contribution measurement. Self-Harness tests current edits for regressions (Zhang et al., 2026a); RELAI-VCL enforces retention during task expansion (Wang et al., 2026a), and EvolveNet validates compositions of independently evolved adaptations (Nie et al., 2026). ACES measures skill value through matched with/without-skill trials (Kevin et al., 2026). These methods assess current competence or realized compositions. LRSI instead prospectively evaluates candidate contributions through paired probes in sampled successors, using standard lower-tail conditional value-at-risk (Rockafellar & Uryasev, 2002) to constrain acceptance; only the selected current edit is committed.

3 Preliminary Study

Does an improvement survive later edits? We retrospectively audit 48 edits accepted by immediate-gain greedy self-improvement on a preliminary ALFWorld split disjoint from the final test set. The edits modify prompt blocks, tool wrappers, retry/validation rules, or routing rules. After acceptance, we generate seven valid successors per edit, yielding 336 edit–successor pairs; these outcomes never affect acceptance or the greedy trajectory. For each pair, we compare the recorded immediate gain Δ_{now} with the retained contribution Δ_{after}, measured by reverting only the earlier edit while holding the successor fixed. Appendix B.8 gives the diagnostic, repeat controls, and statistical definitions.

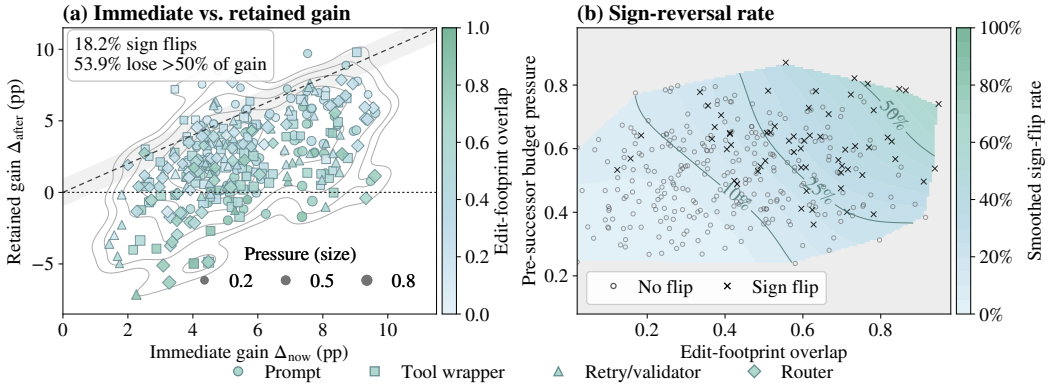


Figure 2: **A locally useful edit need not remain useful after the system is modified again. (a)** Recorded immediate gain versus retained contribution for 336 edit–successor pairs; negative values indicate sign reversals. The dashed diagonal denotes equal gains, with a gray repeat-variation band. **(b)** Smoothed sign-reversal rate versus overlap and budget pressure; gray areas lie outside the observed convex hull. Encoding and smoothing details appear in Appendix B.8.3.

Immediate gains often weaken after composition. Despite positive immediate gains in all pairs, 61/336 (18.2%) have negative retained contributions, and 181/336 (53.9%) retain less than half their immediate gain (Figure 2a). The median paired reduction is 2.83 percentage points, compared with 0.97 points for the 95th percentile of absolute no-successor repeat differences. The mean retained contribution remains positive (2.41 points), so an average alone obscures the harmful minority. Sign-reversal rates also differ across covariate regions: $0/63$ for low overlap and low pressure versus $10/21$ (47.6%) for high overlap and high pressure (Figure 2b; thresholds in Appendix C.1). These rates describe how reversals vary across the observed covariate regions.

Design Principle

Immediate gain can therefore miss both lost contribution and unfavorable successors. This motivates evaluating whether an edit remains useful under plausible later changes and accounting for the lower tail of its retained contribution.

4 METHOD

These findings motivate a selection rule that tests retained contribution in sampled successors.

Overview. An edit that improves an agent can become harmful after a later modification. *Lookahead-Guided Recursive Self-Improvement* (LRSI) first screens edits for immediate improvement, then tests their contribution after possible next edits (Figure 3). For each sampled successor, it measures a candidate's retained contribution by disabling that candidate while keeping the rest of the successor fixed. A lower-tail constraint requires nonnegative mean contribution over the least favorable sampled continuations. LRSI commits only the qualifying edit with the largest immediate gain, then repeats from the updated agent.

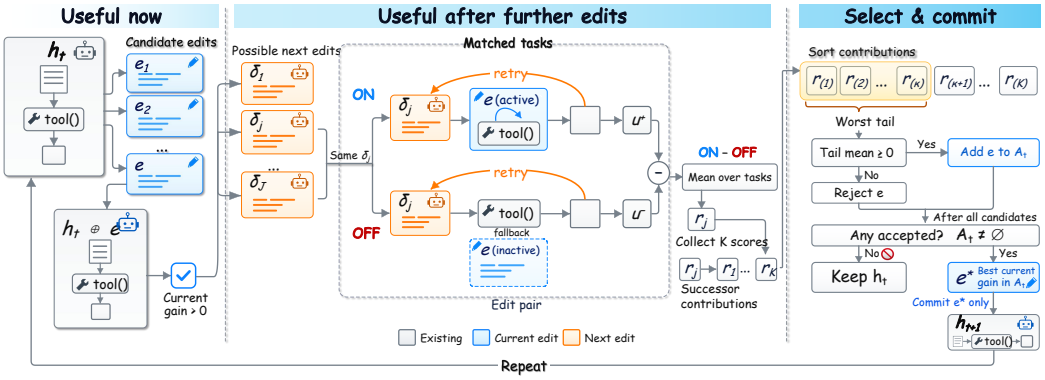


Figure 3: **Overview of Lookahead-Guided Recursive Self-Improvement (LRSI). Left:** Screen candidate edits to the current harness h_t for immediate improvement. **Middle:** For each shortlisted edit e, sample K possible next edits. Hold each δ_j fixed and compare matched task–seed executions with e enabled (ON) and disabled through its registered fallback (OFF). The mean utility difference, r_j in the figure, estimates e's retained contribution. The retry example shows how a useful edit can conflict with a later controller. **Right:** Sort the contributions and accept e if the mean over the worst β fraction of mass is nonnegative. After evaluating all candidates, commit the accepted edit with the largest immediate gain; retain h_t if none qualifies. Discard lookahead edits and repeat.

4.1 SCREENING FOR IMMEDIATE IMPROVEMENT

We freeze the language model M, proposal generator P, and external task verifier, and optimize a harness containing prompt blocks, tool wrappers, and control rules. Within a round, h denotes the fixed current harness h_t. An edit e produces $h_e = h \oplus e$, where $\oplus$ applies a registered modification. For task distribution D, the harness utility and immediate gain are

$$J_D(h) = \mathbb{E}_{x\sim D,\,\omega}[u(h, x, \omega)], \qquad g_{0,D}(e) = J_D(h_e) - J_D(h). \tag{1}$$

Here x is a task, ω is execution randomness, and the verifier returns $u \in [0, 1]$ under a fixed per-task execution limit. We separate generation, development, probe, and test tasks by provenance: tasks connected through shared source identifiers stay in the same split. Only generation traces enter P; development and probe scores guide selection, and test tasks remain sealed until the evolution trajectory is frozen (Appendix A.2).

After evaluating h on a generation batch, we make exactly N independent proposal calls using the same fixed traces. Empty, structurally invalid, and duplicate edits are discarded without replacement, leaving at most N candidates. For $s \in \{\text{gen}, \text{dev}\}$, the paired gain on task–seed batch B_s is $\widehat{g}_{0,s}(e) = |B_s|^{-1} \sum_{(x,\omega)\in B_s} [u(h_e, x, \omega) - u(h, x, \omega)]$. Paired runs start from reset environments with the same task and initial seed. The shortlist $\mathcal{S}_t$ contains the top L candidates by $\widehat{g}_{0,\text{dev}}$ among those satisfying $\widehat{g}_{0,\text{gen}} \geq 0$ and $\widehat{g}_{0,\text{dev}} > 0$. We retain all qualifiers when fewer than L pass; ties follow proposal order.

4.2 MEASURING RETAINED CONTRIBUTION

Consider an edit that adds a retry wrapper, followed by a controller with its own retry loop. Keeping both can exhaust the execution budget even though the wrapper helped the original controller. To isolate the contribution of keeping the wrapper, we compare its enabled and disabled states within the same successor.

Each edit stores immutable code and an interface-compatible parent implementation as its fallback. Disabling e restores that fallback and leaves all other registered code and configuration unchanged; $\mathrm{Off}_e(h \oplus e)$ thus recovers the behavior of h. Later edits must preserve this intervention; an undefined fallback invalidates the comparison. Appendix A.2 specifies the edit contract and a concrete composition example.

For each $e \in \mathcal{S}_t$, run h_e on a fixed generation batch to obtain traces $\mathcal{T}_e$. Each of K continuation slots makes at most A calls to $P(h_e, \mathcal{T}_e; \xi)$ with independent seeds ξ, returning the first nonempty, structurally valid edit. Conditional on structural success, the returned edit has law $q_{P,e}$; repeated valid edits remain repeated samples. A slot with no valid output rejects e. For each returned continuation, construct

$$\delta_j \sim q_{P,e}, \qquad h^+_{e,j} = h_e \oplus \delta_j, \qquad h^-_{e,j} = \mathrm{Off}_e(h^+_{e,j}), \quad j = 1, \dots, K. \tag{2}$$

Both branches keep δ_j's code and configuration fixed; only e's dispatch changes to its registered fallback. An invalid contrast rejects e immediately, without resampling its continuation. Appendix A.3 derives the successful-slot distribution and the effect of this rejection rule.

For fixed h, e, δ_j, the four utilities under a common distribution D form the matrix

$$\mathbf{Y}_D = \begin{bmatrix} J_D(h) & J_D(h^-_{e,j}) \\ J_D(h_e) & J_D(h^+_{e,j}) \end{bmatrix}, \qquad \mathbf{c} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}. \tag{3}$$

The rows represent the candidate disabled and enabled; the columns represent the continuation absent and present. Contrasting the rows gives $\mathbf{c}^\mathsf{T}\mathbf{Y}_D = [g_{0,D}(e), g_{1,D}(e, \delta_j)]$, where $g_{1,D}(e, \delta_j) = J_D(h^+_{e,j}) - J_D(h^-_{e,j})$ is the *retained contribution*. The interaction $I_D(e, \delta_j) = \mathbf{c}^\mathsf{T}\mathbf{Y}_D\mathbf{c} = g_{1,D}(e, \delta_j) - g_{0,D}(e)$ measures how the continuation changes the edit's contribution. A negative interaction weakens the edit; a negative retained contribution makes it harmful inside the successor. Appendix A.4 gives the exact decomposition and its relation to total successor utility.

For selection, we evaluate only the second column on a probe batch $B_{\mathrm{probe}} = \{(x_i, \omega_i)\}_{i=1}^n$:

$$\widehat{r}_{e,j} = \frac{1}{n} \sum_{i=1}^{n} [u(h^+_{e,j}, x_i, \omega_i) - u(h^-_{e,j}, x_i, \omega_i)], \qquad \widehat{\mathbf{r}}_e = (\widehat{r}_{e,1}, \dots, \widehat{r}_{e,K})^\mathsf{T}. \tag{4}$$

Here n counts task–seed pairs; each entry of $\widehat{\mathbf{r}}_e \in [-1, 1]^K$ is the mean contrast for one continuation. Paired runs share tasks and initial seeds, while each branch keeps its own execution distribution after control flow diverges. Task failures receive the verifier's prescribed score; unresolved execution records reject the candidate.

4.3 SELECTING UNDER A LOWER-TAIL CONSTRAINT

An edit can have positive average contribution yet cause large losses under some continuations. LRSI therefore averages the worst β fraction of sampled continuation mass, where $\beta \in (0, 1]$. Equivalently, it minimizes the mean over bounded reweightings:

$$\widehat{R}_\beta(e) = \min_{\mathbf{p} \in \mathcal{P}_{\beta,K}} \mathbf{p}^\mathsf{T}\widehat{\mathbf{r}}_e, \qquad \mathcal{P}_{\beta,K} = \left\{ \mathbf{p} \in [0, (\beta K)^{-1}]^K : \mathbf{1}^\mathsf{T}\mathbf{p} = 1 \right\}. \tag{5}$$

Here $\mathbf{p}$ assigns continuation weights and $\mathbf{1}$ is the all-ones vector. Capping each weight at $1/\beta$ times its uniform value yields empirical lower-tail conditional value-at-risk (CVaR) under a reward convention (Rockafellar & Uryasev, 2002).

To compute the score, sort $\widehat{r}_{e,(1)} \le \cdots \le \widehat{r}_{e,(K)}$, set $\tau = \beta K$, and let $m = \lfloor \tau \rfloor$:

$$\widehat{R}_\beta(e) = \frac{1}{\tau} \left[\sum_{j=1}^{m} \widehat{r}_{e,(j)} + \begin{cases} (\tau - m)\widehat{r}_{e,(m+1)}, & m < K, \\ 0, & m = K. \end{cases} \right]. \tag{6}$$

The sum is empty at $m = 0$; the fractional term handles noninteger tail mass. The score reduces to the ordinary mean at $\beta = 1$ and to the minimum at $\beta \leq 1/K$. Appendix D.1 proves the sorting formula, its dual representation, and its stability to evaluation error.

Let $\mathcal{V}_t \subseteq \mathcal{S}_t$ contain candidates with all K valid, completed contrasts. LRSI accepts candidates with nonnegative tail scores and chooses the largest immediate gain:

$$\mathcal{A}_t = \{e \in \mathcal{V}_t : \widehat{R}_\beta(e) \geq 0\},$$
$$h_{t+1} = \begin{cases} h_t \oplus e_t^*, & e_t^* \in \arg\max_{e \in \mathcal{A}_t} \widehat{g}_{0,\mathrm{dev}}(e), \quad \mathcal{A}_t \neq \varnothing, \\ h_t, & \mathcal{A}_t = \varnothing. \end{cases} \tag{7}$$

Ties follow the same proposal order as screening. A zero empirical tail score is allowed, so useful edits may become redundant in sampled successors. Geometrically, acceptance requires a nonnegative weighted contrast for every $\mathbf{p} \in \mathcal{P}_{\beta,K}$. These homogeneous inequalities define a closed convex cone in $\mathbb{R}^K$; attainable contribution vectors also lie in $[-1, 1]^K$.

Only e_t^* is committed; lookahead edits are discarded, and the next round generates fresh proposals from h_{t+1}. For a shortlist of size L, the paired lookahead evaluations require at most $2LKn$ task executions, and tail computation costs $O(LK \log K)$. Evolution stops at a preset round or resource limit and retains the last completed update; Algorithm 1 and Appendix A.3 detail resource reservation and accounting. This is a local selection rule against harmful retained edits under $q_{P,e}$. We next study robustness to a change in successor distribution; Appendix D.2 addresses estimation error.

4.4 Robust contribution under successor distribution shift

The next accepted update can follow a different distribution from the proposals used for lookahead. To study this shift, fix h, e, D and the generation context, and write $r(\delta) = g_{1,D}(e, \delta) \in [-1, 1]$ and $P = q_{P,e}$. For $\beta \in (0, 1]$, let $R_{\beta,P}$ denote the population lower-tail CVaR of r under P, the counterpart of Equation (5).

Theorem 1 (Robust retained contribution)**.** *Let Q be a successor law for which the same ON/OFF contrast is defined, and let $\varepsilon_\beta = \sup_B\{Q(B) - P(B)/\beta\}$, where B ranges over measurable successor events. Then*

$$\mathbb{E}_Q[r] \geq \max\{-1,\ R_{\beta,P} - 2\varepsilon_\beta\}. \tag{8}$$

In particular, $R_{\beta,P} \geq 0$ guarantees nonnegative expected retained contribution whenever $Q \leq P/\beta$, meaning $Q(B) \leq P(B)/\beta$ for every measurable event B.

Proof sketch. Cap Q at P/β and separate the excess mass ε_β. Allocating the remaining mass to the lowest P-quantiles bounds its contribution from below. Relative to the full β tail, the excess mass costs at most $2\varepsilon_\beta$ because $r \in [-1, 1]$. Appendix D.7 gives the complete derivation. □

Empirical acceptance requires estimation control to invoke this population guarantee (Appendix D.2).

5 Experiments

Our experiments examine whether LRSI improves final performance (**RQ1**, Section 5.2), what the selection criterion contributes (**RQ2**, Section 5.3), when useful edits become harmful (**RQ3**, Section 5.4), and why the lower tail should be inspected (**RQ4**, Section 5.5).

5.1 Experimental setup

Tasks and models. We evaluate on ALFWorld (Shridhar et al., 2021), Terminal-Bench 2.0 (Merrill et al., 2026), SWE-bench Verified (OpenAI, 2024), and Aider Polyglot (Gauthier, 2024), with SWE-bench-Live Lite (Zhang et al., 2025) assessing frozen-harness transfer. We report single-attempt success, pass, and resolved rates. Each condition uses one frozen Qwen3.5-35B-A3B, GPT-5 mini, or Claude Sonnet 4.5 checkpoint for both solving and proposing.

Table 1: **Interactive tasks under matched settings.** ALF: ALFWorld success; TB: Terminal-Bench 2.0 pass rate (%; higher is better). Mean ± sample standard deviation over ten/five evolution seeds, respectively. Em dashes mark benchmarks outside a method's scope. Bold/underlined means denote panel best/second-best; green offsets are gains (pp) over the strongest baseline.

	Qwen3.5-35B-A3B		GPT-5 mini		Claude Sonnet 4.5	
Method	ALF ↑	TB ↑	ALF ↑	TB ↑	ALF ↑	TB ↑
Full-harness optimization						
Fixed ReAct (Yao et al., 2023)	78.36 ± 1.42	24.27 ± 2.19	82.91 ± 1.18	32.47 ± 2.36	86.27 ± 1.07	48.20 ± 2.65
Self-Harness (Zhang et al., 2026a)	84.18 ± 1.36	34.72 ± 2.46	87.16 ± 1.04	40.67 ± 2.20	89.03 ± 0.93	54.04 ± 2.36
Meta-Harness (Lee et al., 2026)	<u>84.93</u> ± 1.31	<u>35.84</u> ± 2.36	<u>87.84</u> ± 1.01	<u>42.13</u> ± 2.28	<u>90.15</u> ± 0.88	<u>56.40</u> ± 2.50
VeRO-Agent Default (Ursekar et al., 2026)	83.73 ± 1.46	34.16 ± 2.56	87.20 ± 1.14	41.35 ± 2.40	89.63 ± 0.98	55.39 ± 2.47
LRSI (ours)	**88.47** ± 1.16 +3.54	**39.33** ± 2.21 +3.49	**91.01** ± 0.90 +3.17	**46.52** ± 2.04 +4.39	**93.10** ± 0.81 +2.95	**60.22** ± 2.23 +3.82
Prompt/skill optimization; executable control flow fixed						
GEPA (Agrawal et al., 2026)	83.54 ± 1.29	<u>31.69</u> ± 2.40	86.60 ± 1.12	<u>38.76</u> ± 2.38	88.66 ± 1.03	<u>51.46</u> ± 2.44
MetaSkill-Evolve (Wang et al., 2026c)	<u>84.48</u> ± 1.34	—	<u>87.28</u> ± 1.08	—	<u>89.37</u> ± 0.97	—
MARS-adapted (Hou et al., 2026)	82.76 ± 1.40	—	85.82 ± 1.17	—	88.02 ± 1.08	—
LRSI (prompt/skill)	**86.90** ± 1.20 +2.42	**35.84** ± 2.30 +4.15	**89.51** ± 0.99 +2.23	**43.15** ± 2.15 +4.39	**91.38** ± 0.89 +2.01	**56.74** ± 2.28 +5.28

Comparison protocol. Baselines span fixed agents, harness optimization, and prompt/skill optimization (Tables 1–2). We match backbones, initial harnesses, information access, and resource limits; prompt/skill panels also match editable components. Generation, development, probe, and test partitions are provenance-disjoint. RQ2–RQ4 use Qwen on ALFWorld (Appendices C.4–C.7). Appendix B details adaptations, splits, hyperparameters, hardware, budgets, and statistical analyses.

5.2 RQ1: Does LRSI improve final performance?

Tables 1–2 compare matched-budget performance within each edit space. Coding agents submit once, without private-test feedback.

Table 2: **Coding tasks under matched settings.** SWE: SWE-bench Verified resolved rate; Poly: Aider Polyglot pass rate (%; higher is better). Mean ± sample standard deviation over five evolution seeds. Matching protocol: Section 5.1; notation: Table 1.

	Qwen3.5-35B-A3B		GPT-5 mini		Claude Sonnet 4.5	
Method	SWE ↑	Poly ↑	SWE ↑	Poly ↑	SWE ↑	Poly ↑
Fixed ReAct (Yao et al., 2023)	36.19 ± 1.91	22.39 ± 1.65	43.20 ± 1.84	27.39 ± 1.79	48.76 ± 1.77	31.74 ± 1.94
Self-Harness (Zhang et al., 2026a)	41.65 ± 1.87	26.96 ± 1.79	47.73 ± 1.69	31.52 ± 1.72	53.09 ± 1.71	35.87 ± 1.72
SICA (Robeyns et al., 2025)	42.68 ± 2.04	27.83 ± 1.82	48.87 ± 1.91	32.39 ± 1.79	54.23 ± 1.80	36.74 ± 1.94
DGM (Zhang et al., 2026b)	44.33 ± 1.96	29.13 ± 1.79	50.62 ± 1.80	34.13 ± 1.82	55.67 ± 1.75	38.26 ± 1.79
HGM (Wang et al., 2026b)	<u>46.49</u> ± 1.84	<u>30.87</u> ± 1.82	<u>52.37</u> ± 1.73	<u>35.65</u> ± 1.79	<u>57.42</u> ± 1.65	<u>39.78</u> ± 1.82
LRSI (ours)	**49.18** ± 1.73	**33.48** ± 1.61	**55.36** ± 1.61	**38.04** ± 1.72	**60.41** ± 1.56	**42.39** ± 1.72

Performance across tasks and edit spaces. Full-harness LRSI exceeds the strongest evaluated baseline in all twelve benchmark–backbone settings by 2.39–4.39 pp. Restricting edits to shared prompt/skill fields also yields gains of 2.01–2.42 pp on ALFWorld and 4.15–5.28 pp on Terminal-Bench.

Transfer and alternative evaluation regimes. The advantage extends to frozen-harness transfer and repository-sharded optimization (Tables 4–5). Source-protocol comparisons also favor LRSI (Tables 6–7); their distinct selection and evaluation rules are detailed in Appendix C.3.

> **Takeaway RQ1.** Selecting for retained contribution can support final performance across full-harness and restricted prompt/skill edits while guiding which changes accumulate.

5.3 RQ2: What does the selection criterion contribute?

Selection controls. Under common resource ceilings and ten paired evolution seeds, LRSI reaches 88.47% success, exceeding all seven controls (Table 8; Appendix B.3). Its advantage over the strongest control, Mean contribution (87.28%), is 1.19 pp (95% CI $[0.31, 2.08]$). Acceptance-matched reaches 85.60% at nearly identical acceptance frequencies (61.0% versus 61.2%), indicating that the selection criterion matters beyond how often edits are accepted.

Independent contribution retention. Figure 4 audits independently confirmed beneficial edits selected from shared candidate pools. LRSI's observed reversal rates are 6.0% for fresh continuations and 10.0% for next-round accepted modifications, versus 19.9% and 28.6% for Current-gain. Rates describe valid contrasts; the next-round comparison has 10–14 per selector and wide marginal intervals. Table 9 provides coverage, noise controls, and contribution means; Appendix C.4 discusses the average–tail tradeoff.

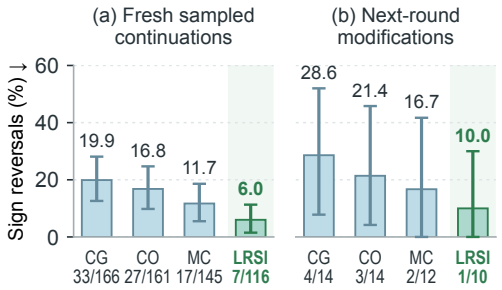


Figure 4: **LRSI has the lowest observed reversal rates.** Bar heights show rates; error bars give clustered 95% CIs. Counts below each bar are reversals/valid contrasts. CG: Current-gain; CO: Compatibility-only; MC: Mean contribution; LRSI: ours. (a) Ten fresh continuation slots per confirmed edit. (b) The next LRSI round's accepted modification. Coverage and noise controls: Table 9.

> **Takeaway RQ2.** Similar resource use and acceptance rates can yield different edits. Independent audits reveal how a selector's choices affect later contributions.

5.4 RQ3: When do useful edits become harmful?

Two audits examine how current and successor gains can conceal harmful edit interactions.

5.4.1 How later edits overturn current gains

Matched four-configuration evaluations decompose retained contribution g_1 into immediate gain g_0 and edit interaction I:

$$I(e, \delta) = g_1(e, \delta) - g_0(e), \qquad g_1 < 0 \iff I < -g_0. \tag{9}$$

Appendices B.9 and C.5 give the protocol and complete statistics, respectively.

Negative interactions affect 89.2% of valid pairs; 230/1,115 (20.6%) cross the reversal boundary (Figure 5). Reversed pairs have a median immediate gain of 4.62 pp: substantial current benefits can be outweighed by later interactions.

5.4.2 How successor gains conceal harmful edits

For 281 shortlisted edits from 98 parents, we compare total successor gain with retained contribution. Given parent h, fixed successor $h^+_{e,\delta}$ with e active, and its counterpart $h^-_{e,\delta}$ with only e disabled,

$$\Delta_{\text{succ}} = J_{\mathcal{D}}(h^+_{e,\delta}) - J_{\mathcal{D}}(h), \qquad g_1 = J_{\mathcal{D}}(h^+_{e,\delta}) - J_{\mathcal{D}}(h^-_{e,\delta}). \tag{10}$$

Appendix B.10 gives the evaluation protocol and gain decomposition.

Of 1,227 improving successor pairs, 167 (13.6%) retain harmful edits (Figure 6). Their median successor gain is $+5.22$ pp and median retained contribution is -1.43 pp, showing how continuation gains can mask harm from the current edit.

> **Takeaway RQ3.** An edit's value depends on the surrounding system. A fixed-successor ON/OFF comparison reveals whether that edit contributes to an improving branch.

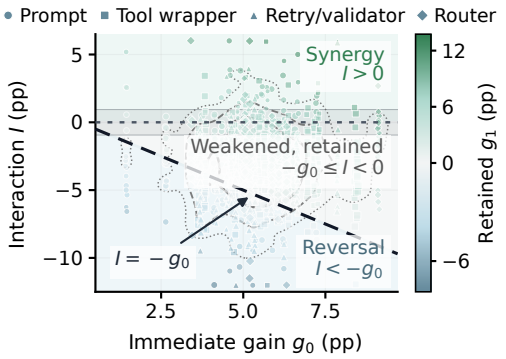


Figure 5: **Negative interactions reverse locally beneficial edits.** Points are edit–continuation pairs; $I = -g_0$ marks the reversal boundary. Color, shape, and area encode retained contribution, edit surface, and budget pressure. Contours show density; the gray band marks repeat noise (Appendix B.9).

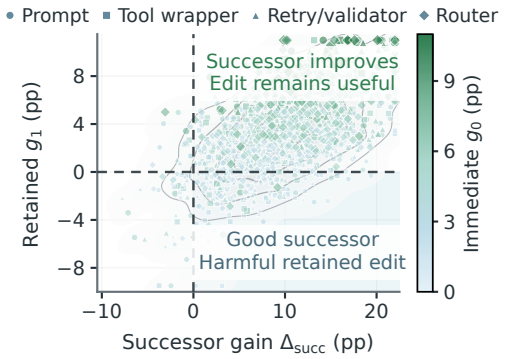


Figure 6: **Positive successor gains can mask negative retained contributions.** Points: edit–continuation pairs; blue quadrant: improving successors with harmful edits. Color/area encode immediate gain; shapes encode edit surface. Contours show relative density; dashed lines mark zero gains (Appendix B.10).

5.5 RQ4: Why inspect the lower tail?

Across 32 fresh continuations per candidate, R_β averages contributions over the worst β fraction of continuation mass; R_1 is the mean (Appendix B.11.1).

Among the 52 complete profiles, 41 have positive means; 15/41 (36.6%) nevertheless have $R_{0.4} < 0$ (Figure 7). Tail-negative and tail-nonnegative groups have similar means (3.08 versus 3.19 pp), with respective reversal rates of 42.9% (6/14) and 9.1% (2/22) after independently accepted next-round modifications (Table 10). Similar mean contributions can therefore conceal different risks of later reversals.

Side strips show negative-contribution frequency, independently measured contribution after an accepted next-round modification, and edit type. Gray marks abstention; the heatmap and contribution strip share a color scale.

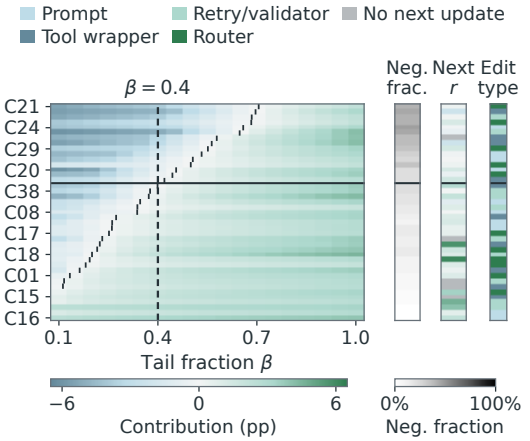


Figure 7: **Positive means conceal negative lower-tail contributions.** Mean-positive edits are sorted by decreasing zero-crossing $\beta^\dagger$ (black). Blue/green show negative/positive contributions; lines mark $\beta = 0.4$ (dashed) and separate tail groups (horizontal).

> **Takeaway RQ4.** A positive average can conceal damaging continuations. A tail constraint limits how much favorable outcomes can compensate for these losses during selection.

6 Conclusion

LRSI selects edits using their retained contribution in prospective successors, with theory characterizing robustness to successor distribution shift. Matched-budget experiments and independent audits show that stronger task performance can accompany fewer observed contribution reversals, while improving successor scores alone can conceal harmful edits. These findings support assessing both agent performance and individual contributions. Future work could adapt the lookahead horizon to edit interactions and examine the accumulation of useful behaviors over longer improvement trajectories.

AI USE STATEMENT

We used generative AI tools to assist in checking mathematical proofs, organizing and presenting experimental results, and improving the clarity and readability of the manuscript. The authors take full responsibility for the final content, including all mathematical claims, experimental results, and conclusions.

REFERENCES

Lakshya A. Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J. Ryan, Meng Jiang, Christopher Potts, Koushik Sen, Alex Dimakis, Ion Stoica, Dan Klein, Matei Zaharia, and Omar Khattab. GEPA: Reflective prompt evolution can outperform reinforcement learning. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/0e9e708b6f48e14fd0ac29e167413f76-Abstract-Conference.html.

Paul Gauthier. o1 tops aider's new polyglot leaderboard, 2024. URL https://aider.chat/2024/12/21/polyglot.html.

Xinmeng Hou, Bohao Qu, Wuqi Wang, Peiliang Gong, Qing Guo, and Yang Liu. Learn like humans: Use meta-cognitive reflection for efficient self-improvement. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 28802–28824. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.1329. URL https://aclanthology.org/2026.acl-long.1329/.

Christopher Kevin, Narendran Raghavan, Jean-Francois Puget, Roshni Malani, Meghana Puvvadi, Moshe Abramovitch, Mohit Gupta, Rama Akkiraju, Subodh Prabhu, Yogesh Dangi, Wei Luo, and Seong Hee Lee. Evaluating skills, not just agents: Agentic continuous evaluation of skills, 2026. URL https://arxiv.org/abs/2608.20614.

Yoonho Lee, Roshen Nair, Qizheng Zhang, Kangwook Lee, Omar Khattab, and Chelsea Finn. Meta-Harness: End-to-end optimization of model harnesses, 2026. URL https://arxiv.org/abs/2603.28052.

Pascal Massart. The tight constant in the Dvoretzky–Kiefer–Wolfowitz inequality. *The Annals of Probability*, 18(3):1269–1283, 1990. doi: 10.1214/aop/1176990746. URL https://doi.org/10.1214/aop/1176990746.

Mike A. Merrill, Alexander G. Shaw, Nicholas Carlini, et al. Terminal-Bench: Benchmarking agents on hard, realistic tasks in command line interfaces, 2026. URL https://arxiv.org/abs/2601.11868.

Jun Nie, Yonggang Zhang, Qianshu Cai, Yiu-ming Cheung, Xinmei Tian, and Bo Han. EvolveNet: Collaborative harness evolution for agent self-improvement, 2026. URL https://arxiv.org/abs/2608.04968.

OpenAI. Introducing SWE-bench Verified, 2024. URL https://openai.com/index/introducing-swe-bench-verified/.

OpenAI. Why we no longer evaluate SWE-bench Verified, 2026. URL https://openai.com/index/why-we-no-longer-evaluate-swe-bench-verified/.

Maxime Robeyns, Martin Szummer, and Laurence Aitchison. A self-improving coding agent, 2025. URL https://arxiv.org/abs/2504.15228.

R. Tyrrell Rockafellar and Stanislav Uryasev. Conditional value-at-risk for general loss distributions. *Journal of Banking & Finance*, 26(7):1443–1471, 2002. doi: 10.1016/S0378-4266(02)00271-6. URL https://doi.org/10.1016/S0378-4266(02)00271-6.

R. Tyrrell Rockafellar, Stan Uryasev, and Michael Zabarankin. Generalized deviations in risk analysis. *Finance and Stochastics*, 10(1):51–74, 2006. doi: 10.1007/s00780-005-0165-8. URL https://sites.math.washington.edu/~rtr/papers/rtr192-GeneralizedDeviations.pdf.

Igal Sason and Sergio Verdú. f-divergence inequalities. *IEEE Transactions on Information Theory*, 62(11):5973–6006, 2016. doi: 10.1109/TIT.2016.2603151. URL https://arxiv.org/abs/1508.00335.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning text and embodied environments for interactive learning. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2010.03768.

Varun Ursekar, Apaar Shanker, Veronica Chatrath, Yuan Xue, and Samuel Marc Denton. VeRO: A harness for agents to optimize agents. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2602.22480.

Wenxiao Wang, Priyatham Kattakinda, and Soheil Feizi. Do agent optimizers compound? a continual-learning evaluation on Terminal-Bench 2.0, 2026a. URL https://arxiv.org/abs/2607.14004.

Wenyi Wang, Piotr Piękos, Nanbo Li, Firas Laakom, Yimeng Chen, Mateusz Ostaszewski, Mingchen Zhuge, and Jürgen Schmidhuber. Huxley-gödel machine: Human-level coding agent development by an approximation of the optimal self-improving machine. In *International Conference on Learning Representations*, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/821d20219c2f14850af1b5220f0ed13f-Paper-Conference.pdf.

Zefeng Wang, Minxi Yan, Jinhe Bi, Sikuan Yan, Volker Tresp, and Yunpu Ma. MetaSkill-Evolve: Recursive self-improvement of LLM agents via two-timescale meta-skill evolution, 2026c. URL https://arxiv.org/abs/2607.05297.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.03629.

Hangfan Zhang, Shao Zhang, Kangcong Li, Chen Zhang, Yang Chen, Yiqun Zhang, Lei Bai, and Shuyue Hu. Self-harness: Harnesses that improve themselves, 2026a. URL https://arxiv.org/abs/2606.09498.

Jenny Zhang, Shengran Hu, Cong Lu, Robert Lange, and Jeff Clune. Darwin gödel machine: Open-ended evolution of self-improving agents. In *International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=pUpzQZTvGY.

Linghao Zhang, Shilin He, Chaoyun Zhang, Yu Kang, Bowen Li, Chengxing Xie, Junhao Wang, Maoquan Wang, Yufan Huang, Shengyu Fu, Elsie Nallipogu, Qingwei Lin, Yingnong Dang, Saravan Rajmohan, and Dongmei Zhang. SWE-bench goes live! In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2025. URL https://arxiv.org/abs/2505.23419.

Appendix Contents

Lookahead-Guided Recursive Self-Improvement Page

A Technical Details . . . 13
A.1 Complete selection procedure . . . 13
A.2 Intervention and evaluation protocol . . . 14
A.3 Proposal law, failure handling, and cost . . . 14
A.4 Exact decomposition of the edit contrast . . . 15
A.5 Applying the population certificate . . . 15

B Experimental Details . . . 16
B.1 Benchmarks, partitions, and access control . . . 16
B.2 Method panels and faithful adaptations . . . 16
B.3 Controls for the selection mechanism . . . 17
B.4 Task performance and mechanism diagnostics . . . 18
B.5 LRSI hyperparameters and resource accounting . . . 19
B.6 Backbones, deployment, and statistical reporting . . . 20
B.7 Original source protocols . . . 20
B.8 Preliminary-study protocol and visualization . . . 21
B.9 Negative-interaction audit protocol . . . 22
B.10 Successor-improvement audit protocol . . . 23
B.11 Dense lower-tail audit protocol . . . 23

C Additional Experimental Results . . . 24
C.1 Preliminary-study statistics . . . 24
C.2 Transfer and sharded comparisons . . . 25
C.3 Comparisons under the original source protocols . . . 26
C.4 Core selection controls and contribution audit . . . 27
C.5 Negative edit-interaction analysis . . . 29
C.6 Successor improvement and retained contribution . . . 29
C.7 Dense lower-tail contribution analysis . . . 29

D Proofs . . . 30
D.1 Tail optimization and geometry . . . 30
D.2 A fixed-candidate audit bound . . . 30
D.3 Quantile allocation and the CVaR risk envelope . . . 31
D.4 Clipping a target distribution against the proposal law . . . 33
D.5 Sharp transfer under successor distribution shift . . . 34
D.6 Density, single-score, and total-variation consequences . . . 35
D.7 Proof of the main-text guarantee . . . 36
D.8 Finite-pool selection: exact transport and its sharp threshold . . . 36
D.9 Composing selection with changed generation evidence . . . 41

A TECHNICAL DETAILS

A.1 COMPLETE SELECTION PROCEDURE

Algorithm 1 uses fixed, nonempty task–seed batches within each round. B_{next} is the generation-split batch used to obtain continuation evidence, and B_{probe} contains n pairs. $\mathsf{Propose}_A$ returns the first structurally valid, nonempty edit within A independent attempts, with $\perp$ denoting exhausted attempts. $\mathsf{Contrast}$ returns the enabled–disabled pair after checking the edit contract, with $\perp$ denoting an inadmissible pair. $\mathsf{PairMean}$ evaluates Equation (4) and returns $\perp$ when execution records remain incomplete after the fixed infrastructure retry policy. All failures and resource use are recorded, including candidates terminated before tail evaluation.

Algorithm 1 LRSI: one-step lookahead for recursive edit selection

Require: Initial harness h_0; frozen M, P and verifier; round-indexed batches $B_{\text{gen}}, B_{\text{dev}}, B_{\text{next}}, B_{\text{probe}}$; round cap T; resource caps C; $N, L, K, A \in \mathbb{N}_{>0}$ and $\beta \in (0, 1]$
Ensure: Final harness and the complete edit, evaluation, and resource log

$h \leftarrow h_0$; save snapshot h
for $t = 0, \ldots, T-1$ **do**
  if the remaining resources cannot reserve a complete round **then**
    break
  end if
  Evaluate h on $B_{\text{gen}}, B_{\text{dev}}$; retain generation traces $\mathcal{T}$
  End evolution if the parent evaluation is incomplete
  Make N independent calls to $P(h, \mathcal{T}; \xi)$; retain distinct, nonempty valid edits $\mathcal{E}$
  Compute $\widehat{g}_{0,\text{gen}}, \widehat{g}_{0,\text{dev}}$; exclude candidates with incomplete evaluations
  $\mathcal{S}_t \leftarrow$ top L edits with $\widehat{g}_{0,\text{gen}} \geq 0$ and $\widehat{g}_{0,\text{dev}} > 0$
  $\mathcal{A}_t \leftarrow \varnothing$
  for $e \in \mathcal{S}_t$ **do**
    $h_e \leftarrow h \oplus e$; obtain traces $\mathcal{T}_e$ on B_{next}
    Skip e if its generation evidence is incomplete
    $\widehat{\mathbf{r}}_e \leftarrow [\,]$
    for $j = 1, \ldots, K$ **do**
      $\delta_j \leftarrow \mathsf{Propose}_A(h_e, \mathcal{T}_e)$ using fresh independent seeds
      if $\delta_j = \perp$ **then**
        Log proposal failure; **break**
      end if
      $\mathcal{H} \leftarrow \mathsf{Contrast}(h, e, \delta_j)$
      if $\mathcal{H} = \perp$ **then**
        Log contrast failure; **break**
      end if
      $\widehat{r}_{e,j} \leftarrow \mathsf{PairMean}(\mathcal{H}, B_{\text{probe}})$
      if $\widehat{r}_{e,j} = \perp$ **then**
        Log incomplete evaluation; **break**
      end if
      Append $\widehat{r}_{e,j}$ to $\widehat{\mathbf{r}}_e$
    end for
    if $|\widehat{\mathbf{r}}_e| = K$ **then**
      Compute $\widehat{R}_\beta(e)$ by Equation (6)
      If $\widehat{R}_\beta(e) \geq 0$, add e to $\mathcal{A}_t$
    end if
  end for
  if $\mathcal{A}_t \neq \varnothing$ **then**
    Choose $e_t^* \in \arg\max_{e \in \mathcal{A}_t} \widehat{g}_{0,\text{dev}}(e)$, breaking ties by proposal order
    $h \leftarrow h \oplus e_t^*$
  end if
  Save snapshot h and the round log
end for
return h and logs

A.2 Intervention and evaluation protocol

Registered edits. An edit records its identifier, owned implementation, content hash, dependencies, and fallback at the parent state. The fallback preserves the interface exposed to subsequent edits. Later modifications add versioned implementations and update registered routes while preserving earlier owned code. To construct a contrast, copy the successor and change only the candidate's dispatch to its fallback. Both branches retain identical continuation code, configurations, and remaining dispatch rules. The contract requires well-defined dependencies in both branches and forbids regenerating continuation code during intervention. Construction checks compare ownership hashes, interfaces, and dependency resolution, followed by answer-independent smoke tests. The admissible space is restricted to registered edits satisfying this contract; arbitrary unregistered rewrites are rejected.

A two-edit example. Suppose the parent sends each tool request through a single-attempt wrapper. Candidate e replaces that wrapper with a bounded retry loop and stores the single-attempt implementation as its fallback. Continuation δ installs a controller that calls the same wrapper interface inside an outer retry loop. In h^+, both retry loops are active; in h^-, the same controller calls the single-attempt fallback. The controller's code, its retry limit, and its call sites remain fixed across the pair. Thus a negative contrast attributes a utility loss to retaining the inner loop within that controller. If a continuation bypasses the wrapper completely, disabling e leaves execution unchanged and its population contribution is zero. If a continuation requires an interface unavailable under the fallback, the pair is inadmissible and the candidate is rejected.

Task grouping and paired execution. For each available provenance category—source instance, environment, repository—add an undirected edge between tasks sharing an identifier. Groups are the connected components of the resulting graph; each group is assigned to exactly one of the generation, development, probe, and test splits before evolution. The benchmark-specific identifier mappings and split lists determine this partition. Each round fixes its task–seed batches before proposals are generated. Parent generation traces are reused for current proposals; continuation proposals use only traces from h_e on B_{next}. Development and probe observations remain outside proposal prompts. These selection splits can be reused across rounds; final test observations enter only the evaluation of the frozen trajectory.

Each paired execution starts from a clean environment and uses the same task and initial seed, with the verifier and per-task limits held fixed. The execution engine preserves the marginal randomness of each branch, including after control-flow divergence. Ordinary task failures and timeouts receive the prescribed utility. Infrastructure failures follow a fixed retry policy shared across methods; an unresolved record makes the corresponding candidate ineligible. Successful low-scoring runs are never retried to obtain a replacement score.

A.3 Proposal law, failure handling, and cost

Fix h_e and its generation traces $\mathcal{T}_e$. Let $q_{0,e}$ be the raw proposal law and S_e the set of nonempty, structurally valid continuations. Write $s_e = q_{0,e}(S_e)$. For $s_e > 0$, the successful-slot distribution is $q_{P,e}(E) = q_{0,e}(E \cap S_e)/s_e$ for every measurable set E. With an attempt cap A, the slot returns failure with probability $(1 - s_e)^A$ and returns an edit in E with probability $[1 - (1 - s_e)^A]q_{P,e}(E)$. This follows by summing over the first successful attempt, whose index a has probability $(1 - s_e)^{a-1}s_e$. When $s_e = 0$, the slot always fails and $q_{P,e}$ is unused. Independent seeds and fixed traces make successful slots independent before subsequent contrast checks. Duplicates are retained, preserving the law $q_{P,e}$.

A structurally successful proposal is tested once for an admissible contrast. Failure at this stage rejects the candidate immediately, without another proposal attempt. Every continuation slot must complete before the tail score is computed. For independent slot-validity events with common probability v_e, passing all K slots has probability v_e^K. Consequently, K controls both sampling resolution and the strength of validity screening. The probability v_e characterizes validity under the fixed protocol. Logs retain proposal exhaustion, inadmissible contrasts, and missing outcomes as distinct failure types.

Resource accounting. Let $n_g = |B_{\text{gen}}|$, $n_d = |B_{\text{dev}}|$, and $n_s = |B_{\text{next}}|$; the probe batch has size n. Excluding infrastructure retries, one round uses at most $C_{\text{runs}} \le (N+1)(n_g+n_d)+Ln_s+2LKn$ task executions. The parent is evaluated once per current batch, and its stored outcomes are paired with every candidate. There are exactly N current proposal calls in a round that reaches proposal generation and at most ALK continuation calls. Tail computation uses $O(LK \log K)$ time.

The resource caps C specify ceilings on generated tokens, task executions, and elapsed time; every proposal attempt, build check, execution, and infrastructure retry is charged to the applicable counters. Before a round, resource reservation uses the maximum output lengths, task time limits, and retry caps. Evolution stops when a full round cannot be reserved, subject also to the round cap T. An interrupted round contributes its actual cost but commits no candidate. Deployment executes the final harness without generating lookahead branches.

A.4 Exact decomposition of the edit contrast

Fix h, e, δ, D and let y_{ab} denote the utility at candidate activation $a \in \{0,1\}$ and continuation activation $b \in \{0,1\}$. Suppressing the fixed arguments, define $g_0 = y_{10} - y_{00}$, $g_\delta = y_{01} - y_{00}$, and $I = y_{11} - y_{10} - y_{01} + y_{00}$. The four utilities satisfy the exact identity $y_{ab} = y_{00} + ag_0 + bg_\delta + abI$. Equivalently, with $\mathbf{v} = (0,1)^\mathsf{T}$ and $\mathbf{1}_2 = (1,1)^\mathsf{T}$,

$$\mathbf{Y}_D = y_{00}\mathbf{1}_2\mathbf{1}_2^\mathsf{T} + g_0\mathbf{v}\mathbf{1}_2^\mathsf{T} + g_\delta\mathbf{1}_2\mathbf{v}^\mathsf{T} + I\mathbf{v}\mathbf{v}^\mathsf{T}. \tag{11}$$

The identities $\mathbf{c}^\mathsf{T}\mathbf{1}_2 = 0$ and $\mathbf{c}^\mathsf{T}\mathbf{v} = 1$ give $g_1 = g_0 + I$ and $\mathbf{c}^\mathsf{T}\mathbf{Y}_D\mathbf{c} = I$. The expansion holds for every four-entry utility table, without assuming that program behavior is linear. All four configurations use the same D; an interaction audit therefore evaluates both columns on a common reserved task batch.

Total successor improvement satisfies $J_D(h^+) - J_D(h) = g_\delta + g_1$. It can be large even when retaining e is harmful. For example, let the parent utility be 0.4 and the immediate gain be 0.1. A successor with $g_\delta = 0.5$ and $g_1 = -0.05$ has utility 0.85; one with $g_\delta = 0.05$ and $g_1 = 0.1$ has utility 0.55. Both tables obey the utility bounds, yet a tail determined by the former rejects its negative retained contribution. LRSI's constraint therefore selects for contribution retention within the successor. Since δ is generated from h_e and held fixed during intervention, the contrast also excludes changes in the continuation-generation process caused by accepting e.

A.5 Applying the population certificate

The population guarantees in Appendix D assume that the ON/OFF contrast is defined on the relevant supports. A successor with zero proposal probability but a valid contrast is covered through singular mass in Equation (39). Undefined contrasts remain outside the reward domain and are tracked as uncovered probability mass. If one conditions on identifiable accepted successors, the acceptance probability and target law in Theorem 4 must use that same event, and all excluded probability must be reported separately.

The results also distinguish population positivity from empirical acceptance. For example, if an independent audit establishes $|\widehat{R}_\beta - R_{\beta,P}| \le \xi$ on an event of probability at least $1-\eta$, Corollary 1 gives, for $C = \alpha/\beta \ge 1$,

$$R_{\alpha,Q} \ge \max\left\{-1,\ \widehat{R}_\beta - \xi - \frac{2\varepsilon_C(Q\|P)}{\alpha}\right\} \quad \text{on the same audit event.} \tag{12}$$

Applying this inequality requires a valid audit error bound that accounts for the evaluation data's independence or adaptive reuse. The empirical rule $\widehat{R}_\beta \ge 0$ supplies a selection score; certifying nonnegative population contribution additionally requires covering the estimation and distribution-shift penalties. Appendix D.2 states the fixed-candidate audit assumptions and quantifies the required estimation margin. The certificate describes the marginal effect of e within a fixed successor; evaluating e by total successor utility or multi-round value uses a separate objective.

B EXPERIMENTAL DETAILS

This section defines the benchmark partitions, comparison methods, implementation configuration, and statistical analyses for evaluating LRSI.

B.1 BENCHMARKS, PARTITIONS, AND ACCESS CONTROL

Benchmark coverage. ALFWorld (Shridhar et al., 2021) provides six families of text-based household tasks. We reserve its 134 unseen evaluation games for final testing and build the optimization partitions from its training games, grouping by the underlying scene identifier. Terminal-Bench 2.0 (Merrill et al., 2026) contributes 89 terminal tasks; we use three prespecified outer folds, with related source projects/task variants assigned together and task categories balanced where possible. Every task is tested only by a harness evolved without its outer fold, and cross-fold aggregates weight tasks equally. The matched evaluation pool comprises all 89 tasks; the separate Self-Harness source-protocol comparison in Table 6 uses a filtered 64-task setting. Deployment feasibility is checked before optimization, with a common task manifest across methods.

For repository repair, SWE-bench Verified supplies 500 human-screened instances (OpenAI, 2024); we partition by repository and keep the benchmark's official resolution tests. Aider Polyglot supplies 225 exercises across six programming languages (Gauthier, 2024); translations or variants of the same underlying exercise remain together. The final test partitions contain 97 SWE-bench Verified instances and 46 Polyglot exercises, shared by all methods and backbones; centralized and repository-sharded SWE comparisons use the same 97 instances. Following the DGM/HGM coding evaluation, private correctness tests and their results are unavailable to the solving agent, and each execution produces one final submission. Evaluation scores are specific to this single-submission protocol; Aider's two-attempt setting exposes test feedback between submissions.

SWE-bench-Live Lite contains 300 tasks in the original release (Zhang et al., 2025). It serves only as an external transfer test for harnesses evolved on SWE-bench Verified, with no Live-based search or checkpoint selection. Removing exact issue duplicates leaves 277 tasks (All), of which 203 belong to repositories absent from optimization (New). These fixed evaluation sets are shared by all methods and backbones. This benchmark measures transfer to a different task collection. Pretraining exposure and verifier reliability remain relevant to interpreting these results (OpenAI, 2026).

Partition construction. We use deterministic group assignment with split seed 2026. Within ALFWorld's training pool and each Terminal-Bench outer training pool, target proportions are $60/20/20$ for generation, development, and probe tasks. For SWE-bench Verified and Polyglot, target proportions are $40/20/20/20$ for generation, development, probe, and final test. Whole-group assignment takes precedence over exact proportions. The split manifest records actual counts, category/language balance, and group identifiers. No task migrates between roles within a run. All methods share the same manifests. Test-fold identity is never an optimizer input, and no hyperparameter is selected by comparing outer-fold test results.

Permitted feedback. Only generation tasks may provide trajectories, failure messages, or demonstrations to an optimizer. Development and probe feedback is restricted to task identifiers and scalar verifier scores for selection; test tasks remain inaccessible until every evolution trajectory and analysis choice is frozen. Baselines may allocate their scalar-selection queries across the development and probe pools under the same total budget. These adaptively queried pools serve as selection data; final-test data provide the held-out evaluation. Benchmark source files, private tests, solution artifacts, and test logs are isolated from the optimizer process. Public tests ordinarily available inside a coding repository remain available uniformly to all solvers.

B.2 METHOD PANELS AND FAITHFUL ADAPTATIONS

Full harness optimization. The common initial agent uses a fixed ReAct-style interaction loop (Yao et al., 2023), with the same prompt, environment actions, file-editing/shell tools where applicable, and per-task limits. The unchanged agent is the no-optimization reference. Self-Harness (Zhang et al., 2026a), Meta-Harness (Lee et al., 2026), and VeRO-Agent Default (Ursekar et al., 2026) are compared with LRSI on ALFWorld and Terminal-Bench, using benchmark adapters

where necessary. Self-Harness retains its native regression gate and compatible-edit merging; Meta-Harness retains its archive and history access; VeRO-Agent retains its optimizer/tool configuration. History access respects the same generation-trace and scalar-selection boundary. VeRO denotes infrastructure, so the comparison row identifies the concrete optimizer, *VeRO-Agent Default*.

The SWE-bench Verified and Polyglot panel compares the unchanged agent, LRSI, Self-Harness, SICA (Robeyns et al., 2025), DGM (Zhang et al., 2026b), and HGM (Wang et al., 2026b). This selection follows HGM's Table 2, which directly compares SICA, DGM, and HGM on both coding benchmarks. We retain each method's archive, parent-selection rule, and utility/budget allocation policy, including HGM's clade-based search. Initial task-solving behavior and available tools are matched, while method-specific optimizer state remains intact. These controlled reruns use the matched model roles, optimization subsets, and evaluation access specified here.

EvolveNet (Nie et al., 2026) appears in an additional SWE-bench Verified panel. Its local-search/aggregation structure uses up to five clients with repository-grouped generation shards and a shared scalar-selection service. The client count is capped by the number of available generation groups. Both methods operate under this access regime: an explicitly labeled *LRSI-sharded* adapter allocates its eight fixed-proposer calls across shards in a deterministic rotation, without exchanging raw traces. Every client's generation, aggregation, and evaluation cost counts toward one global budget. This panel evaluates optimization performance under repository-sharded access separately from centralized LRSI.

Prompt and skill optimization. GEPA (Agrawal et al., 2026), a reflective prompt optimizer also used in VeRO's Table 2, is compared on ALFWorld and Terminal-Bench. MetaSkill-Evolve (Wang et al., 2026c), whose Table 1 includes ALFWorld, is included on ALFWorld with its fast task-skill and slow meta-skill loops retained. MARS (Hou et al., 2026), published at ACL 2026 for QA, is evaluated on ALFWorld as *MARS-adapted*. We adapt its category-conditioned metacognitive reflection to the six ALFWorld task families using generation traces only. For each of these comparisons, LRSI may edit the identical solver prompt/skill fields, with tool implementations and executable control flow fixed. Optimizer-specific memory and reflection remain method-specific and are fully charged. These restricted LRSI results are separated from the full-harness panel.

Comparison design. All panels freeze model weights and the external verifier. LRSI additionally freezes its proposal procedure; methods that evolve their own optimization policy may retain that capability in full-system comparisons. Thus the full-system panel evaluates total optimization performance, while the shared-proposal controls below isolate LRSI's selection rule. The primary comparison uses frozen-weight harness optimization and single-attempt evaluation with external verifiers. Published discovery-set scores from Meta-Harness and VeRO follow separate evaluation protocols.

B.3 Controls for the selection mechanism

The mechanism study uses the full-harness Qwen3.5-35B-A3B/ALFWorld condition, with the same initial agent, 134 test games, ten evolution seeds, and resource ceilings as Table 1. It tests whether retained-contribution selection improves complete recursive optimization beyond immediate gain, additional search resources, successor utility, structural compatibility, mean contribution, or a lower acceptance frequency. The seven controls isolate individual choices in the selection procedure.

Current-gain. Use LRSI's generation/development screen and commit the largest development gain without lookahead.

Extra-Eval. Spend the saved lookahead allocation on new task–seed evaluations of current candidates, aggregate their immediate gains, and select the best candidate satisfying the same screen. This tests whether repeated evaluation alone explains an improvement.

Extra-Candidates. Spend that allocation on additional proposals and their immediate evaluations, charging both generation and evaluation costs. It shares the first eight proposals with LRSI and expands that pool using the saved allocation.

Two-step lookahead. Rank candidates by the mean absolute successor utility of LRSI's sampled continuations, $J(h \oplus e \oplus \delta)$. Commit only e, keeping the deployed update to one edit. In

recursive runs, resources saved by omitting OFF executions may fund additional continuations or repetitions under the common budget.

Compatibility-only. Enforce all continuation build/interface/fallback checks, then rank by immediate gain without the retained-contribution gate. This separates executable compatibility from measured utility contribution.

Mean contribution. Replace the lower-tail gate by the mean over all K retained contributions, retaining LRSI's final ranking rule.

Acceptance-matched. Raise the current-gain selector's development-gain threshold to match LRSI's acceptance frequency on pilot data, then freeze it. Realized acceptance rates are also reported; no test-based adjustment is allowed.

All seven controls undergo complete recursive optimization (Table 8). Proposal templates, generation-trace access, task schedules, initial random seeds, and tie-breaking are matched; different commitments can lead to different trajectories and later candidates. Extra-Candidates shares the first eight proposals at a common parent and charges every additional proposal and evaluation. The table reports optimization task executions, generated tokens including exposed reasoning tokens, and accepted/attempted rounds to quantify search resources and acceptance frequency.

The fixed-parent audit compares Current-gain, Compatibility-only, Mean contribution, and LRSI on the same candidate pools and paired batches. For each seed, its parents are the starts of rounds 1 and 5 of the Current-gain trajectory, chosen without test scores. An unreached round remains a missing parent in the coverage record. All four selectors use the same generation/development screen and deterministic shortlist; the latter three share the same K selection continuations and validity checks. Their selected edits, including abstentions, define the audit cohort. Identical parent–edit pairs selected by more than one rule share audit executions and remain paired in analysis. The resulting comparison characterizes the edits selected by each policy.

B.4 Task performance and mechanism diagnostics

Primary outcomes. Let $u(h, x, s) \in \{0, 1\}$ be the native verifier outcome of one reset execution. We report

$$\widehat{J}_{\text{test}}(h) = \frac{1}{|D_{\text{test}}|R} \sum_{x \in D_{\text{test}}} \sum_{s=1}^{R} u(h, x, s), \qquad R = 2. \tag{13}$$

ALFWorld uses task success, Terminal-Bench and Polyglot use pass rate, and both SWE-bench datasets use resolved rate. The two independent executions are averaged to estimate single-attempt success. For Table 1, the percentage rate for evolution seed i is $p_i = 100c_i/268$ on ALFWorld and $p_i = 100c_i/178$ on Terminal-Bench, where c_i is the integer number of successful executions. Terminal-Bench first pools the three disjoint outer folds with task-count weights within each seed, then aggregates the five seed-level rates. Across five evolution seeds, the reported means for SWE-bench Verified, Polyglot, Live All, and Live New aggregate 970, 460, 2,770, and 2,030 binary outcomes, respectively; standard deviations are computed across the five seed-level rates. Tables 1, 2, and 5 report $\bar{p} \pm s$, with sample standard deviation $s = \sqrt{\sum_{i=1}^{n}(p_i - \bar{p})^2/(n-1)}$: $n = 10$ for ALFWorld and $n = 5$ for the other benchmarks in these tables. Means and standard deviations are computed from unrounded seed-level rates and rounded only for display. Public tool interaction within an execution is permitted; feedback from the final private verifier is not returned for a repair attempt. Timeouts, agent exceptions, and invalid final outputs count as failures.

Final evaluation uses the last deployable harness when optimization stops. Test scores never select checkpoints, hyperparameters, or acceptance thresholds. The main benchmark panels concern endpoint performance under common resource ceilings.

Independent contribution audit. After freezing the Qwen/ALFWorld runs, a fresh test-only execution batch confirms the current gain of each selected edit. An edit enters the contribution summary only if its paired 95% confidence lower bound exceeds a nonnegative tolerance ϵ, fixed from independent pilot repeat/no-op noise. A second independent execution batch measures $g_0(e)$ and $r(e, \delta)$ using matched task–seed pairs and reset environments. Audit task manifests and repetition counts are fixed before inspecting these outcomes. Table 9 reports selected/available parents, confirmed/selected edits, intervention coverage, mean retained contribution, and the fraction of valid

contrasts with a negative point estimate. This descriptive sign-reversal rate uses the number of negative point estimates over valid contrasts. A smaller positive contribution can reflect replacement by a later edit while retaining a benefit.

For each confirmed edit, panel A uses ten fresh continuation slots generated from generation-only context, independent of the K selection slots. Panel B executes one complete next LRSI selection round from the same candidate branch and audits the modification it actually accepts. The next-round policy, resource limits, and access boundary are common across all four current-edit selectors. In both panels, only e is disabled; successor code and configuration remain fixed. Undefined or incomplete interventions and next-round no-update outcomes retain separate coverage counts and are excluded from contribution summaries without replacement. Panel B evaluates accepted modifications in these controlled branches. Audit executions are outside the search budget and never feed back into selection.

Panel C summarizes fresh noise controls on the same audit cohort: a no-op continuation yields the contrast $r(e, \delta_{\text{noop}}) - g_0(e)$, while two independent executions of an unchanged harness yield a repeat-score difference. Fresh audit repetitions bypass execution-result caching. The mean contrast and 95th percentile of its absolute magnitude quantify audit noise, with ϵ fixed throughout. These mechanism experiments use Qwen/ALFWorld with fixed proposal and generation settings.

B.5 LRSI HYPERPARAMETERS AND RESOURCE ACCOUNTING

The defaults are $N = 8$, $L = 3$, $K = 5$, $\beta = 0.4$, $A = 3$, and retained-contribution threshold zero. Thus the empirical lower tail averages the two smallest valid contributions when all five slots are available. Structural continuation failures follow Appendix A.3: each slot has at most three generation attempts; repeated valid continuations retain their multiplicity, while undefined/incomplete contrasts invalidate the candidate and are never replaced by favorable samples. Empty, duplicate, or invalid current proposals are removed without replacement. Only the selected current edit is committed; sampled continuations are discarded.

Each round draws eight distinct tasks from each optimization role and uses two execution repetitions, giving $n_g = n_d = n_s = n = 16$ task–seed pairs in the cost notation of Appendix A.3. If a group split supplies fewer than eight tasks, its full pool is used and the size disclosed. Role-specific task schedules are shared across methods and rotate deterministically between rounds. Parent and branch executions use matched environment seeds and independent reset states. For hosted APIs without reproducible seeded sampling, pairing uses request blocks and repeated draws; token-level randomness follows each provider's sampling behavior.

The default per-round upper bound, before infrastructure retries, is

$$C_{\text{round}} \le (N+1)(n_g + n_d) + Ln_s + 2LKn = 816. \tag{14}$$

There are exactly eight current-proposal calls and at most $ALK = 45$ continuation-generation calls, in addition to the solver calls inside these task executions. LRSI uses at most ten rounds on ALFWorld and five on each other optimization benchmark. All methods receive a common task-execution ceiling of $8{,}160$ or $4{,}080$, respectively; baselines may use more optimization rounds within the same ceiling. For Terminal-Bench this ceiling applies independently to each outer fold. The unchanged reference has zero optimization cost.

A separate deployment pilot fixes the generated-token and elapsed-time ceilings, with identical limits for every method within a benchmark/backbone condition. Each method stops at the first exhausted cap. Runs reserve the worst-case resources for their next indivisible evaluation/selection unit; only a completed unit can authorize a commit. Invalid edits, abandoned branches, subprocesses, subagents, and retries all consume budget. At most one retry is permitted for a logged infrastructure failure, using the same request and charging both attempts; ordinary agent failures receive their verifier score after the first attempt. An unresolved infrastructure failure is reported in coverage and counts as unsuccessful in the primary intention-to-evaluate score. Exact-request caching is allowed uniformly and its hits are logged. The execution ledger separates optimization from audit and final-test calls; diagnostic results never return to the optimizer.

Hyperparameters are fixed at the defaults above; the seven named selection controls are the component comparisons. Noninteger βK uses the fractional boundary weight in the Method. Hyperparam-

eters and the pilot-derived ϵ are locked before final-test access. The five sampled successors supply the empirical selection score; population guarantees use the estimation conditions in Appendix D.2.

B.6 Backbones, deployment, and statistical reporting

Model roles. The three backbones are Qwen3.5-35B-A3B, GPT-5 mini, and Claude Sonnet 4.5, used by Self-Harness, HGM's evaluator, and VeRO's optimizer, respectively. In our reruns, one checkpoint supplies the solver, optimizer, and subagents within each condition. Model weights, verifier, and LRSI's proposal template remain fixed. We record revision hashes or dated API identifiers, serving configuration, reasoning mode, and provider version before execution. Replacing an unavailable snapshot requires a separately labeled condition and rerunning all paired methods.

Deployment. The reference deployment specifies four H200 GPUs for Qwen, matching the GPU count/type reported by Self-Harness, and hosted inference for the other two backbones. The environment host configuration specifies 64 CPU cores and 256 GB RAM, with at most eight concurrent task containers for all methods. The run manifest records GPU memory, CPU model, driver/runtime versions, provider region, and prices. Concurrency and resource ceilings are fixed within each comparison; resource accounting includes both total CPU/GPU-hours and elapsed time, charging parallel search for its full compute cost.

Inference and environment limits. All methods share a 128k-token context ceiling, up to 8,192 generated tokens per call including billed reasoning tokens where exposed, and the same truncation policy. Trajectory ceilings are 32,768 generated tokens and 50 environment actions for ALFWorld, and 65,536 tokens and 100 tool turns for coding/terminal tasks. Terminal-Bench retains task-specific native timeouts; SWE-bench and Polyglot use a one-hour limit per execution. The earliest applicable limit terminates an execution. Qwen's solver temperature is 0.6 and proposal temperature is 1.0, with top-$p = 0.95$; each is shared across methods. API backbones use one provider-supported decoding/reasoning configuration per role, frozen after the pilot. Every reasoning setting, hidden-token accounting limitation, tool version, container digest, prompt, and network/dependency policy is disclosed. Environment installation and resource mirroring are completed uniformly before timed solving.

Repetitions and inference. The evaluation uses ten independent evolution seeds on ALFWorld and five on each terminal/coding benchmark, paired across methods; Terminal-Bench uses five per outer fold. Final evaluation uses the two fresh executions in Eq. (13); benchmark tables report seed means and standard deviations. For the core recursive controls, Δ is LRSI minus the row method, in percentage points, with a paired 95% hierarchical-bootstrap interval obtained by resampling whole evolution runs and task-provenance groups. For Table 9, audit intervals preserve run/parent clustering and shared parent–edit records across selectors. The separate interaction analysis uses the parent-cluster interval specified in Appendix B.9.1. Continuations (K per candidate), edits from one parent, and repeated solver calls remain grouped within their evolution runs. The intervals summarize uncertainty without multiplicity adjustment. Run records retain seeds, task manifests, budget usage, accepted/attempted rounds, audit coverage, and stopped or infrastructure-affected runs.

B.7 Original source protocols

Self-Harness protocol. Self-Harness uses a filtered 64-task Terminal-Bench 2.0 pool and the same Qwen3.5-35B-A3B model for solving and proposing, starting from its minimal DeepAgent harness (Zhang et al., 2026a). Figure 4 reports pass rates aggregated over two repeated evaluations. Its "held-out" split participates in promotion and is therefore labeled *regression gate*. The implementation uses four H200 GPUs and 48 concurrent task executions.

HGM protocol. HGM compares SICA, DGM, and HGM from a shared DGM-derived ancestor with an allowance of 800 task evaluations (Wang et al., 2026b). SWE-Verified-60 uses GPT-5 for expansion and GPT-5 mini for execution. Polyglot uses Qwen3-Coder-480B-A35B-Instruct (mixed int4/int8) and Qwen3-Coder-30B-A3B-Instruct for these roles. These are selected-agent scores under the source protocol, distinct from our final-test results. Published initial-agent scores come from Section 4.2 and published baseline results from Table 2.

B.8 PRELIMINARY-STUDY PROTOCOL AND VISUALIZATION

B.8.1 RETROSPECTIVE PROTOCOL AND PAIRED DIAGNOSTIC

The preliminary study uses an ALFWorld split disjoint from the final test set. The starting trajectories use immediate-gain greedy self-improvement, accepting edits solely on their current gains. The edit space is restricted in advance to four modular surfaces—prompt blocks, tool wrappers, retry/validation rules, and routing rules—for which an isolated post-hoc toggle can be constructed. We collect 48 accepted edits and, only after each is accepted, independently generate seven valid later edits from its updated agent. These 336 edit–successor pairs form a retrospective audit, separate from the greedy selection process.

For record i, let h_i be the agent before current edit e_i, let δ_i be its sampled successor, and let $S_{\mathcal{B}_i}$ be the average verifier score on the audit batch, fixed within that paired comparison. We record

$$\Delta_{\mathrm{now},i} = S_{\mathcal{B}_i}(h_i \oplus e_i) - S_{\mathcal{B}_i}(h_i), \tag{15}$$

and construct a diagnostic successor copy in which only e_i is reverted:

$$\Delta_{\mathrm{after},i} = S_{\mathcal{B}_i}(h_i \oplus e_i \oplus \delta_i) - S_{\mathcal{B}_i}\big((h_i \oplus e_i \oplus \delta_i)^{-e_i}\big)\,. \tag{16}$$

The later edit δ_i stays fixed during reversion. A negative retained contribution means that keeping e_i hurts this successor relative to reverting it, even if the successor improves on the original agent. A smaller positive contribution can reflect replacement or redundancy. We call a negative retained contribution a sign reversal.

The unit of analysis is an edit–successor pair. Immediate gain and pre-successor budget pressure vary across the seven pairs sharing an edit, so Equations 15–16 retain each pair's values with index i. The audit batch is fixed within a pair and may vary across records. The 336 pairs share 48 edits; these 336 observations form clusters at the edit level.

B.8.2 REPEAT CONTROLS AND COVARIATES

Each accepted edit also has five repeated evaluations without a successor, providing 240 control records. For each control, the repeat difference is the repeated gain minus that control's recorded reference gain. We pool the absolute repeat differences and take their empirical 95th percentile, using linear interpolation between adjacent ordered values. Its value is 0.97 percentage points and defines the half-width of the gray band around $\Delta_{\mathrm{after}} = \Delta_{\mathrm{now}}$ in Figure 2a. This descriptive band shows the scale of repeat variation.

Two recorded covariates characterize the pairs. *Edit-footprint overlap* is the weighted Jaccard overlap of modules/interfaces touched by the current and later edits: the total weight of their shared footprint divided by that of their union. *Pre-successor budget pressure* is the fraction of the allowed execution budget consumed by the current edited agent before the later edit is introduced. Both lie in $[0, 1]$ and enter only the retrospective descriptive analysis. The low-overlap/low-pressure group uses strict thresholds $\mathrm{overlap} < 0.35$ and $\mathrm{pressure} < 0.50$; the high-overlap/high-pressure group uses $\mathrm{overlap} > 0.60$ and $\mathrm{pressure} > 0.65$. These groups illustrate two covariate regions within the full set of pairs.

All gain summaries in Appendix C.1 are computed over the 336 records. A sign reversal is $\Delta_{\mathrm{after},i} < 0$; losing more than half the immediate gain is $\Delta_{\mathrm{after},i} < 0.5\Delta_{\mathrm{now},i}$. Gains are multiplied by 100 to express score differences in percentage points (pp). In particular, the median paired reduction is computed from the within-record differences.

B.8.3 FIGURE CONSTRUCTION

In Figure 2a, each marker is one audited pair. Shape identifies the current-edit surface (circle: prompt; square: tool wrapper; triangle: retry/validation; diamond: router), the color scale encodes footprint overlap, and marker area increases linearly with budget pressure. The dashed diagonal denotes equal immediate and retained contributions, and the dotted horizontal line marks zero retained contribution. The gray density contours come from a bivariate Gaussian kernel density estimate with bandwidth factor 0.28. Density is evaluated on a 160×160 grid over the displayed axes; contour levels are the 10th, 30th, 50th, 70th, and 90th percentiles of grid densities above their 55th percentile. The contours show relative pair density.

For Figure 2b, let o_i and b_i be the recorded overlap and pressure, and $z_i = \mathbf{1}\{\Delta_{\text{after},i} < 0\}$. The displayed rate at (o, b) is the Gaussian-kernel weighted fraction

$$\widehat{p}(o,b) = \frac{\sum_i w_i(o,b) z_i}{\sum_i w_i(o,b)}, \qquad w_i(o,b) = \exp\left[-\frac{1}{2}\left(\frac{(o-o_i)^2}{0.12^2} + \frac{(b-b_i)^2}{0.11^2}\right)\right]. \tag{17}$$

We evaluate it on a 140×140 grid with overlap from 0.02 to 0.98 and pressure from 0.08 to 0.98, masking locations outside the observed covariates' convex hull. The color scale spans 0–100%; contour lines mark 10%, 25%, and 50% where those levels occur. Open circles mark non-negative retained contributions and crosses mark sign reversals. The convex-hull mask restricts the surface to the sampled geometric range, within which local observation density varies. Table 3 reports the raw subgroup counts.

B.9 Negative-interaction audit protocol

We use the full-harness Qwen3.5-35B-A3B/ALFWorld condition in Appendix B.3. The starts of rounds 1 and 5 of each of ten Current-gain trajectories define 20 prespecified parents. At each parent, we retain three candidates from the common generation/development screen before applying retained-contribution selection, yielding 60 edits.

Each edit receives $K_{\text{mech}} = 20$ fresh continuation slots drawn from generation-only context, independently of optimization continuations. We freeze each valid continuation and evaluate the registered ON/OFF intervention on a common fresh task–seed batch B_{mech}, with matched initial seeds and independent environment resets. All evaluations are post-hoc. Undefined or incomplete interventions are recorded without replacement and excluded from numerical summaries. The 1,200 attempted pairs yield 1,115 valid contrasts and 85 undefined or incomplete pairs (92.9% coverage).

B.9.1 Four-cell interaction and uncertainty

Let $h^+_{e,\delta}$ be the frozen successor with edit e active and $h^-_{e,\delta} = \text{Off}_e(h^+_{e,\delta})$ the same successor with only e disabled. The four utilities on the common batch are

$$\begin{aligned} y_{00} &= J(h), & y_{10} &= J(h \oplus e), \\ y_{01} &= J(h^-_{e,\delta}), & y_{11} &= J(h^+_{e,\delta}). \end{aligned} \tag{18}$$

In percentage-point units, the immediate gain, retained contribution, and interaction are

$$\begin{aligned} g_0(e) &= y_{10} - y_{00}, & g_1(e,\delta) &= y_{11} - y_{01}, \\ I(e,\delta) &= y_{11} - y_{10} - y_{01} + y_{00}, & g_1 &= g_0 + I. \end{aligned} \tag{19}$$

Holding the continuation fixed in both successor configurations gives the exact reversal boundary

$$g_1(e,\delta) < 0 \quad \Longleftrightarrow \quad I(e,\delta) < -g_0(e). \tag{20}$$

Figure 5 locates audited pairs relative to this boundary.

We weight valid pairs equally and define reversal by $g_1 < 0$. For the reversal-rate interval, 3,000 parent-cluster bootstrap replicates retain all edits and continuations at each sampled parent together. Their 2.5th and 97.5th percentiles give the 95% interval $[17.7, 23.4]\%$.

The gray band spans ± 0.94 pp around $I = 0$. This repeat-noise scale is the 95th percentile of the absolute unchanged-harness repeat difference in Table 9C.

Figure construction. Statistics and density contours use the four utility measurements. Display coordinates apply opposite offsets of at most 0.025 pp to g_0 and I to separate coincident points while preserving g_1, the interaction sign, and the reversal region. A two-dimensional Gaussian KDE with bandwidth factor 0.22 supplies the approximate 50%, 75%, and 90% highest-density contours. Marker area increases with recorded pre-successor budget pressure; shape identifies prompt, tool-wrapper, retry/validator, or router edits.

B.10 SUCCESSOR-IMPROVEMENT AUDIT PROTOCOL

We audit frozen Qwen3.5-35B-A3B/ALFWorld trajectories from ten runs. The cohort comprises 281 shortlisted edits from 98 parents: 85 parents contribute three edits and 13 contribute two. Five fixed continuation slots per edit yield 1,405 attempted pairs, of which 1,281 provide valid contrasts (91.2% coverage). The 124 undefined or incomplete pairs are excluded from numerical summaries.

For each valid pair, we evaluate the parent h and the matched successors $h^+_{e,\delta} = h \oplus e \oplus \delta$ and $h^-_{e,\delta} = \mathrm{Off}_e(h^+_{e,\delta})$ on a common fresh task–seed batch, with identical initial seeds, verifier, and execution limits. The two successors share the continuation code and configuration; only the dispatch of e switches to its registered fallback. The audit leaves candidate selection and the frozen trajectories unchanged.

B.10.1 GAIN DECOMPOSITION

The total successor gain and the retained contribution measure different contrasts:

$$\Delta_{\mathrm{succ}}(e,\delta) = J_{\mathcal{D}}(h^+_{e,\delta}) - J_{\mathcal{D}}(h), \tag{21}$$

$$g_1(e,\delta) = J_{\mathcal{D}}(h^+_{e,\delta}) - J_{\mathcal{D}}(h^-_{e,\delta}). \tag{22}$$

Defining $g_\delta(e,\delta) = J_{\mathcal{D}}(h^-_{e,\delta}) - J_{\mathcal{D}}(h)$ gives

$$\Delta_{\mathrm{succ}}(e,\delta) = g_\delta(e,\delta) + g_1(e,\delta). \tag{23}$$

For $g_1 < 0$, the successor improves precisely when $g_\delta > -g_1$: the gain without e outweighs the loss from retaining it. The fixed-successor intervention isolates this retained contribution.

Figure construction. Figure 6 weights valid pairs equally. A two-dimensional Gaussian KDE with bandwidth factor 0.23 provides the relative-density background and contours. Color and area encode immediate gain g_0; shape distinguishes prompt, tool-wrapper, retry/validator, and router edits. The blue quadrant marks $\Delta_{\mathrm{succ}} > 0$ and $g_1 < 0$.

B.11 DENSE LOWER-TAIL AUDIT PROTOCOL

We evaluate the complete deterministic shortlists at the 20 prespecified parents from Appendix B.9. Each parent contributes up to $L = 3$ candidates, for at most 60 edits. The audit leaves the evolution trajectories and selection rule unchanged.

For each structurally admissible candidate, we draw $K_{\mathrm{audit}} = 32$ fresh continuations independently of the selector's five continuations. Each ON/OFF contrast $r_{e,j}$ averages repeated batches of 32 matched task–seed pairs, with the continuation implementation, verifier, and execution limits fixed. This yields 52 complete profiles and 1,664 continuation-level contrasts in percentage points.

B.11.1 TAIL PROFILES AND INDEPENDENT NEXT-ROUND AUDIT

For each complete profile $\mathbf{r}_e = (r_{e,1}, \ldots, r_{e,32})^\mathsf{T}$, we compute

$$R_\beta(e) = \min_{\mathbf{p} \in \mathcal{P}_{\beta,32}} \mathbf{p}^\mathsf{T}\mathbf{r}_e, \qquad \beta \in \{0.10, 0.15, \ldots, 1.00\}, \qquad R_1(e) = \bar{r}_e, \tag{24}$$

using the fractional-boundary convention in Equation (6). Within the mean-positive cohort ($R_1 > 0$), we distinguish tail-negative ($R_{0.4} < 0$) and tail-nonnegative ($R_{0.4} \geq 0$) candidates. Figure 7 orders rows by decreasing

$$\beta^\dagger(e) = \inf\{\beta \in (0,1] : R_\beta(e) \geq 0\}. \tag{25}$$

The crossing is computed with fractional boundary mass; larger values indicate that more favorable continuations are needed to offset the adverse tail. The grayscale strip reports $32^{-1}\sum_j \mathbf{1}\{r_{e,j} < 0\}$.

Each mean-positive candidate branch undergoes one independent complete next LRSI round. We measure retained contribution after an accepted modification on fresh matched batches, independently of the tail-profile continuations. Negative contrasts are reversals. The five abstentions are shown in gray and excluded from reversal-rate denominators.

C Additional Experimental Results

Bold and underlined values indicate the best and second-best results within each column, following the indicated metric direction.

C.1 Preliminary-study statistics

Table 3: Descriptive summaries of the preliminary audit. Gain differences are in percentage points; subgroup rates use their own denominators.

Quantity	Value
Sign reversals	$61/336$ (18.2%)
Retained contribution below half the immediate gain	$181/336$ (53.9%)
Median immediate gain	5.24 pp
Median retained contribution	2.30 pp
Median paired reduction	2.83 pp
Mean retained contribution	2.41 pp
95th percentile of absolute repeat differences	0.97 pp
Sign reversals: low overlap and low pressure	$0/63$ (0.0%)
Sign reversals: high overlap and high pressure	$10/21$ (47.6%)

The high-overlap/high-pressure group has a higher observed reversal rate and contains 10 of the 61 reversals. Reversals also occur across a broader range of overlap: among the 84 pairs above 0.60, 51 have non-negative retained contributions, while 28 reversals occur among the 252 pairs at or below 0.60. Overlap and pressure thus describe variation in reversal risk, with both beneficial and harmful pairs occurring in the same covariate ranges.

This descriptive audit characterizes contribution loss among modular, reversible edits selected by immediate gain. It motivates prospective contribution measurement and the controlled evaluations of LRSI that follow.

C.2 TRANSFER AND SHARDED COMPARISONS

Table 4 evaluates the harness selected on SWE-bench Verified without further optimization on SWE-bench-Live Lite. The two columns per backbone distinguish the deduplicated evaluation pool (All; 277 tasks) from its repository-disjoint subset (New; 203 tasks). Table 5 isolates repository-sharded optimization with a single global resource budget and the same scalar-selection interface. These comparisons retain the checkpoint and evaluation rules of Appendix B. For matched runs, scores are mean ± standard deviation over evolution seeds; averaging the two solver executions per task estimates single-attempt performance. Terminal-Bench outer-fold scores are task-weighted before seed-level aggregation. Appendix C.4 reports the mechanism controls and their paired uncertainty estimates.

The frozen LRSI harness exceeds HGM by 1.87–2.35 pp on the All and New partitions of SWE-bench-Live Lite. Under repository-sharded optimization, LRSI-sharded exceeds EvolveNet by 2.78–2.89 pp.

Table 4: **Frozen-harness transfer to SWE-bench-Live Lite.** Resolved rate (%, ↑) after evolution on SWE-bench Verified. All: 277 tasks remaining after issue deduplication; New: the 203-task subset with repositories absent from optimization. No Live-based selection or adaptation is allowed. Results are mean ± standard deviation over evolution seeds.

	Qwen3.5-35B-A3B		GPT-5 mini		Claude Sonnet 4.5	
Method	All ↑	New ↑	All ↑	New ↑	All ↑	New ↑
Fixed ReAct (2023)	31.73 ± 1.76	28.92 ± 1.91	38.16 ± 1.69	35.07 ± 1.85	43.29 ± 1.66	40.10 ± 1.78
Self-Harness (2026a)	35.92 ± 1.72	32.61 ± 1.87	41.88 ± 1.63	38.42 ± 1.79	47.18 ± 1.60	43.50 ± 1.74
SICA (2025)	36.64 ± 1.81	33.15 ± 1.95	42.71 ± 1.72	39.01 ± 1.88	48.01 ± 1.68	44.24 ± 1.82
DGM (2026b)	37.83 ± 1.77	34.29 ± 1.90	43.90 ± 1.67	40.15 ± 1.84	49.21 ± 1.64	45.32 ± 1.79
HGM (2026b)	39.13 ± 1.71	35.62 ± 1.86	45.27 ± 1.61	41.38 ± 1.78	50.43 ± 1.59	46.40 ± 1.74
LRSI (ours)	**41.37 ± 1.65**	**37.49 ± 1.81**	**47.62 ± 1.55**	**43.35 ± 1.72**	**52.74 ± 1.53**	**48.28 ± 1.70**

Table 5: **Repository-sharded optimization on SWE-bench Verified.** Resolved rate (%, ↑) on the same 97 test instances as Table 2, under a common global budget. EvolveNet and LRSI-sharded share the client partition and information access; the fixed agent is the common initial reference. Results are mean ± sample standard deviation over five evolution seeds. This panel is separate from centralized optimization.

Method	Qwen3.5-35B-A3B	GPT-5 mini	Claude Sonnet 4.5
Fixed ReAct (2023)	36.19 ± 1.91	43.20 ± 1.84	48.76 ± 1.77
EvolveNet (2026)	44.12 ± 1.88	50.31 ± 1.73	55.26 ± 1.73
LRSI-sharded (ours)	**47.01 ± 1.80**	**53.09 ± 1.67**	**58.14 ± 1.65**

C.3 COMPARISONS UNDER THE ORIGINAL SOURCE PROTOCOLS

Tables 6 and 7 retain the original evaluation conditions of Self-Harness and HGM. Values marked † are the published point estimates from those papers. Unmarked LRSI rows report our results under the respective source protocols. These scores and costs are reported under their source protocols, separately from the held-out comparisons in Section 5.2. Appendix B.7 specifies the source evaluation conditions.

Table 6: **Terminal-Bench under the Self-Harness source protocol.** Qwen3.5-35B-A3B; pass rate (%, ↑) on the source's 64-task pool. †: published Figure 4 values from Zhang et al. (2026a). The regression-gate column is the source's "held-out" column and is used during optimization.

Method	Held-in	Regression gate	Overall
Initial DeepAgent†	15.1	23.8	18.0
Self-Harness†	<u>36.0</u>	<u>38.1</u>	<u>36.7</u>
LRSI (source protocol)	**39.5**	**42.9**	**40.6**

Table 7: **Coding tasks under the HGM source protocol.** †: values from Wang et al. (2026b), including their SICA/DGM reruns. Acc.: accuracy (%); CPU-h: allocated CPU-hours. Polyglot uses the source evaluation pool.

	SWE-Verified-60		Polyglot (source pool)	
Method	Acc. ↑	CPU-h ↓	Acc. ↑	CPU-h ↓
Initial agent†	40.0		20.3	
SICA†‡ (Robeyns et al., 2025)	50.0	*stalled*	25.4	572
DGM† (Zhang et al., 2026b)	53.3	1,231	27.1	2,385
HGM†	<u>56.7</u>	<u>517</u>	<u>30.5</u>	<u>347</u>
LRSI (source protocol)	**59.2**	**465**	**33.1**	**326**

‡ SICA stalls after 360 SWE-Verified-60 evaluations (context-window errors; source time: "infinite loop"). CPU-hours are hardware/accounting dependent.

C.4 CORE SELECTION CONTROLS AND CONTRIBUTION AUDIT

The core study separates complete recursive performance (Table 8) from contributions measured at common parents (Table 9). Both use Qwen3.5-35B-A3B on ALFWorld under Appendix B.3.

Search resources and lookahead target. LRSI reaches 88.47% task success, compared with 85.37% for Extra-Eval and 86.12% for Extra-Candidates. Ranking by total successor utility reaches 85.75%, while checking structural compatibility alone reaches 85.19%. Mean contribution is the strongest control at 87.28%; LRSI's paired advantage is 1.19 pp (95% CI $[0.31, 2.08]$). Acceptance-matched reaches 85.60% at an acceptance frequency of 61.0%, close to LRSI's 61.2%. Together, these controls support measuring retained contribution under adverse continuations beyond increasing search effort, checking compatibility, or scoring overall successor quality.

Table 8: **Recursive selection controls.** Qwen3.5-35B-A3B on ALFWorld; ten paired evolution seeds and the same resource ceilings as Table 1. Success is final single-attempt task success (%, mean $\pm$ SD). Δ is LRSI minus the row method (pp), with a paired 95% confidence interval. Counts and tokens include optimization only. An em dash marks the self-comparison.

Selector	Success ↑	Δ [95% CI]	Accepted a/t (%)	Executions	Gen. tokens (M)
Current-gain	84.89 ± 1.47	$+3.58$ [2.54, 4.62]	$90/100$ (90.0%)	2,731	8.94
Extra-Eval	85.37 ± 1.36	$+3.10$ [2.19, 4.01]	$87/99$ (87.9%)	7,906	22.87
Extra-Candidates	86.12 ± 1.31	$+2.35$ [1.37, 3.31]	$90/98$ (91.8%)	8,021	24.36
Two-step lookahead	85.75 ± 1.40	$+2.72$ [1.64, 3.82]	$82/99$ (82.8%)	7,725	23.11
Compatibility-only	85.19 ± 1.44	$+3.28$ [2.47, 4.09]	$84/100$ (84.0%)	3,205	10.47
Mean contribution	$\underline{87.28} \pm 1.24$	$+1.19$ [0.31, 2.08]	$71/98$ (72.4%)	7,832	23.56
Acceptance-matched	85.60 ± 1.34	$+2.87$ [1.83, 3.92]	$61/100$ (61.0%)	2,718	8.81
LRSI (ours)	$\mathbf{88.47} \pm 1.16$	—	$60/98$ (61.2%)	7,865	23.82

Task executions and generated tokens are averaged over evolution seeds; a/t aggregates accepted/attempted rounds over those seeds. Limits apply per run. Generated tokens include exposed reasoning tokens for solver and proposer calls, including failed attempts. Acceptance-matched uses a pilot-fixed threshold. Bold and underlined values rank success means only, marking the best and second-best results.

Average contribution and reversal risk. In the fresh audit, Mean contribution yields a slightly higher average retained contribution than LRSI (3.21 versus 3.09 pp) and a higher reversal rate (11.7% versus 6.0%). The lower-tail gate targets adverse continuations that the average can conceal. LRSI also retains positive mean contribution under next-round accepted modifications (2.63 pp; 95% CI $[1.14, 4.13]$). These summaries describe valid contrasts from independently confirmed edits in the selected cohorts; the next-round comparison has 10–14 valid contrasts per selector and wide marginal intervals.

Table 9: **Independent contribution audit at common parents.** Qwen3.5-35B-A3B on ALFWorld. Panel A evaluates ten fresh sampled continuations per confirmed edit. Panel B evaluates the modification accepted by one complete next LRSI round from that edit's branch. The four current-edit selectors share parent/candidate pools. Retained contribution $\bar{r}$ is in pp; contribution means and sign-reversal rates include clustered 95% confidence intervals. Reversal rates count negative point estimates and describe the audited cohorts.

A. Fresh sampled continuations

Current selector	Selected a/P	Confirmed b/a	Coverage $v/u/z$	$\bar{r}\uparrow$ [95% CI]	Reversals $\downarrow k/v$ (%) [CI]
Current-gain	20/20	18/20	166/14/0	2.27 [1.47, 3.08]	33/166 19.9% [12.6, 28.1]
Compatibility-only	19/20	17/19	161/9/0	2.42 [1.68, 3.16]	27/161 16.8% [9.8, 24.7]
Mean contribution	16/20	15/16	145/5/0	**3.21** [2.43, 3.99]	17/145 <u>11.7</u>% [5.5, 18.6]
LRSI (ours)	13/20	12/13	116/4/0	<u>3.09</u> [2.31, 3.87]	7/116 **6.0**% [1.5, 11.3]

B. Modification accepted by the next LRSI round

Current selector	Selected a/P	Confirmed b/a	Coverage $v/u/z$	$\bar{r}\uparrow$ [95% CI]	Reversals $\downarrow k/v$ (%) [CI]
Current-gain	20/20	18/20	14/1/3	1.31 [−0.12, 2.76]	4/14 28.6% [7.8, 52.0]
Compatibility-only	19/20	17/19	14/1/2	1.58 [0.19, 2.96]	3/14 21.4% [4.2, 45.8]
Mean contribution	16/20	15/16	12/1/2	<u>2.18</u> [0.74, 3.61]	2/12 <u>16.7</u>% [0.0, 41.7]
LRSI (ours)	13/20	12/13	10/0/2	**2.63** [1.14, 4.13]	1/10 **10.0**% [0.0, 30.0]

C. Independent noise controls on the audit cohort

Control	Paired records	Mean (pp) [95% CI]	95th $\lvert\text{contrast}\rvert$ (pp)
No-op: $r(e, \delta_{\text{noop}}) - g_0(e)$	25	+0.03 [−0.17, 0.22]	0.78
Unchanged-harness repeat difference	25	−0.06 [−0.29, 0.18]	0.94

P: available prespecified parents; a: selected edits; b: independently confirmed beneficial edits. Coverage lists valid (v), undefined/incomplete (u), and no-update (z) slots among confirmed edits. Thus $v + u + z = 10b$ in A (where $z = 0$) and $v + u + z = b$ in B. Reversal counts use only the v valid contrasts; $v = 0$ gives an undefined rate. The same a/P and b/a apply across A and B. Noise controls deduplicate shared parent–edit records. Contribution summaries include only valid contrasts from confirmed edits. Within each panel, bold and underlined point estimates mark the best and second-best $\bar{r}$ and reversal rate; these rankings are descriptive.

C.5 NEGATIVE EDIT-INTERACTION ANALYSIS

Of 1,115 valid pairs, 995 (89.2%) have negative interaction: 765 weaken the current edit while retaining positive contribution, and 230 (20.6% of all valid pairs) reverse its contribution. The remaining 120 pairs have positive interactions. Among reversed pairs, 206/230 (89.6%) have immediate gains of at least 3 pp. Median immediate gains are 4.62 and 5.58 pp for reversal and non-reversal pairs; the corresponding median interactions are -6.13 and -2.47 pp, and median retained contributions are -1.33 and 3.11 pp. Each median is computed separately over the relevant pair cohort.

Thus, negative interactions usually weaken an edit while preserving its benefit, but sufficiently large interaction losses reverse even substantial immediate gains. Successor-conditioned evaluation measures this loss directly through the paired ON/OFF contrast.

C.6 SUCCESSOR IMPROVEMENT AND RETAINED CONTRIBUTION

Of 1,281 valid pairs, 1,060 have both positive successor gains and positive retained contributions, while 167 have positive successor gains and negative retained contributions. Among the 54 pairs with negative successor gains, 16 have positive and 38 have negative retained contributions. No valid contrast lies on either zero boundary.

The 167 pairs with $\Delta_{\text{succ}} > 0$ and $g_1 < 0$ constitute 13.6% of all improving successor pairs. Within this group, median successor gain is $+5.22$ pp and median retained contribution is -1.43 pp. Moreover, 46/167 pairs (27.5%) gain more than 8 pp over the parent, showing that the mismatch extends to successors with substantial improvements.

C.7 DENSE LOWER-TAIL CONTRIBUTION ANALYSIS

Table 10 reports candidate-weighted group statistics. Of 41 mean-positive edits, 15 are tail-negative and 26 have strictly positive $R_{0.4}$. The independent next round accepts modifications for 14 and 22 candidates, respectively, with six and two reversals. Similar mean contributions therefore coexist with distinct lower-tail profiles and next-round reversal rates.

Table 10: Dense audit of mean-positive edits. Group means weight candidates equally; reversal rates are computed over accepted next-round modifications.

Statistic	$R_{0.4} < 0$	$R_{0.4} \geq 0$
Candidates	15	26
Mean R_1 (pp)	3.08	3.19
Mean $R_{0.4}$ (pp)	-2.47	$+1.37$
Accepted next modifications	14	22
No next update	1	4
Next-round reversals	6/14 (42.9%)	2/22 (9.1%)

D PROOFS

D.1 TAIL OPTIMIZATION AND GEOMETRY

For $\mathbf{r} \in \mathbb{R}^K$, define $\rho_\beta(\mathbf{r}) = \min_{\mathbf{p} \in \mathcal{P}_{\beta,K}} \mathbf{p}^\mathsf{T}\mathbf{r}$. In particular, $\widehat{R}_\beta(e) = \rho_\beta(\widehat{\mathbf{r}}_e)$.

Proposition 1 (Tail representation). *For $0 < \beta \le 1$, ρ_β equals the sorted expression in Equation (6) and has the dual representation*

$$\rho_\beta(\mathbf{r}) = \max_{z \in \mathbb{R}} \left\{ z - \frac{1}{\beta K} \sum_{j=1}^{K} (z - r_j)_+ \right\}, \qquad (w)_+ = \max\{w, 0\}. \tag{26}$$

Here z is the optimization variable in the dual representation.

Proof. Order the entries increasingly and set $\tau = \beta K$. If $r_i < r_j$, $p_i < 1/\tau$, and $p_j > 0$, moving mass from j to i decreases the objective while preserving feasibility. An optimum therefore assigns mass $1/\tau$ to the first $\lfloor \tau \rfloor$ entries and the remaining mass to the next entry. For $\tau < 1$, all mass is assigned to a minimum entry; for $\tau = K$, every weight equals $1/K$. This gives Equation (6), including fractional tail sizes and ties.

Attach a multiplier z to the equality $1 - \sum_j p_j = 0$. Minimizing each Lagrangian term over $0 \le p_j \le 1/\tau$ gives $\inf_{0 \le p_j \le 1/\tau} p_j(r_j - z) = -(z - r_j)_+/\tau$. The primal linear program is feasible and bounded, so strong duality yields Equation (26), the lower-tail CVaR representation (Rockafellar & Uryasev, 2002). □

Proposition 2 (Acceptance geometry and stability). *The set $\mathcal{C}_{\beta,K} = \{\mathbf{r} \in \mathbb{R}^K : \rho_\beta(\mathbf{r}) \ge 0\}$ is a closed convex cone. For $0 < \beta_1 \le \beta_2 \le 1$, it satisfies $\mathcal{C}_{\beta_1,K} \subseteq \mathcal{C}_{\beta_2,K}$. Moreover, $|\rho_\beta(\mathbf{r}) - \rho_\beta(\mathbf{s})| \le \|\mathbf{r} - \mathbf{s}\|_\infty$ for all $\mathbf{r}, \mathbf{s} \in \mathbb{R}^K$.*

Proof. The acceptance set is the intersection of the homogeneous closed half-spaces $\{\mathbf{r} : \mathbf{p}^\mathsf{T}\mathbf{r} \ge 0\}$ over $\mathbf{p} \in \mathcal{P}_{\beta,K}$. Reducing β enlarges this family of constraints, establishing the inclusion. For any feasible $\mathbf{p}$, $|\mathbf{p}^\mathsf{T}(\mathbf{r} - \mathbf{s})| \le \|\mathbf{r} - \mathbf{s}\|_\infty$ because $\mathbf{p}$ is a probability vector. Applying this inequality at a minimizer for each vector proves stability. □

The bounded acceptance region is $\mathcal{C}_{\beta,K} \cap [-1, 1]^K$; it is convex but generally not a cone. The stability bound transfers a uniform error bound on paired means to their tail score, including dependent errors across continuations.

D.2 A FIXED-CANDIDATE AUDIT BOUND

Fix a parent, candidate, and generation evidence, and write $r(\delta) = g_{1,D}(e, \delta) \in [-1, 1]$ for the true retained contribution on D. For a continuation law q, define

$$R_{\beta,q} = \frac{1}{\beta} \int_0^\beta F_q^{-1}(v)\, dv = \max_{z \in [-1,1]} \left\{ z - \frac{1}{\beta} \mathbb{E}_{\delta \sim q}[(z - r(\delta))_+] \right\}. \tag{27}$$

Here F_q is the cumulative distribution function of $r(\delta)$ and F_q^{-1} is its generalized quantile function. The integral counts exactly β probability mass, including a fractional contribution from a quantile atom. The bound below compares the proposal law q_P with a fixed target law q_{sel}, such as the law of an update accepted by the next complete selection round. The candidate index is suppressed throughout this subsection.

Theorem 2 (Audit error decomposition). *Suppose $\delta_1, \ldots, \delta_K$ are independent draws from q_P and the contrast is defined on the supports of both q_P and q_{sel}. Each $\widehat{r}_j$ averages n independent task–seed pairs from D, independent of the fixed candidate, its generation evidence, and the sampled continuations; both branches have their correct marginal execution laws. The same task–seed batch can be shared across continuations. Let $d = \mathrm{TV}(q_P, q_{\mathrm{sel}}) = \sup_E |q_P(E) - q_{\mathrm{sel}}(E)|$ and $0 < \eta < 1$. With probability at least $1 - \eta$,*

$$|\widehat{R}_\beta - R_{\beta,q_{\mathrm{sel}}}| \le \underbrace{\sqrt{\frac{2\log(4K/\eta)}{n}}}_{\text{task evaluation}} + \underbrace{\frac{2}{\beta}\sqrt{\frac{\log(4/\eta)}{2K}}}_{\text{continuation sampling}} + \underbrace{\frac{2d}{\beta}}_{\text{distribution mismatch}}. \tag{28}$$

Proof. Let $\mathbf{r}^\circ = (r(\delta_1), \ldots, r(\delta_K))^\mathsf{T}$ and let $\widehat{F}_K$ be its empirical distribution function. Each paired outcome is in $[-1, 1]$. Hoeffding's inequality followed by a union bound gives $\Pr(\|\widehat{\mathbf{r}} - \mathbf{r}^\circ\|_\infty > \epsilon) \leq 2K \exp(-n\epsilon^2/2)$. Setting $\epsilon = \sqrt{2\log(4K/\eta)/n}$ makes this probability at most $\eta/2$; Proposition 2 transfers the same error to ρ_β.

The Dvoretzky–Kiefer–Wolfowitz inequality (Massart, 1990) gives $\|\widehat{F}_K - F_{q_P}\|_\infty \leq \alpha$ with probability at least $1 - \eta/2$, where $\alpha = \sqrt{\log(4/\eta)/(2K)}$. For a random contribution $R \in [-1, 1]$ and $z \in [-1, 1]$, the identity $\mathbb{E}[(z - R)_+] = \int_{-1}^{z} F_R(s)\, ds$ bounds the hinge-expectation error by 2α. Taking the maximum in Equation (27) therefore bounds the continuation-sampling error by $2\alpha/\beta$.

Finally, $(z - r(\delta))_+ \in [0, 2]$, so total variation bounds the difference between its expectations under q_P and q_{sel} by $2d$. Maximizing over z yields $|R_{\beta,q_P} - R_{\beta,q_{\text{sel}}}| \leq 2d/\beta$. The triangle inequality and the two probability bounds establish the result. □

The theorem applies to a fixed candidate evaluated on fresh, independent audit data under the stated sampling law. Adaptive reuse of selection probes and filtering of invalid contrasts require corresponding dependence and sampling adjustments. Target continuations without defined interventions fall outside the estimand; their probability mass is recorded as uncovered. A no-update outcome is included as $\delta = \varnothing$, for which $r(\varnothing) = g_{0,D}(e)$.

The three terms separate task noise, finite continuation sampling, and transition mismatch. Increasing n and K reduces the first two terms but leaves the third unchanged. For example, at $K = 5$, $\beta = 0.4$, and $\eta = 0.05$, the continuation term alone is approximately 3.31, exceeding the maximum possible score difference of 2. At this sampling scale the bound exceeds the score range. The empirical threshold $\widehat{R}_\beta \geq 0$ guides selection; a population certificate requires enough audit samples to control the estimation margin.

D.3 Quantile allocation and the CVaR risk envelope

All measures below are finite measures on a common measurable space $(\Omega, \mathcal{F})$. An inequality $\mu \leq \nu$ means $\mu(B) \leq \nu(B)$ for every $B \in \mathcal{F}$. The reward $r : \Omega \to [-1, 1]$ is the retained contribution of the same fixed candidate under the same task distribution. We write $P = q_{P,e}$ for the probe law and condition on the fixed parent, candidate, and generation context. The analysis allows discrete rewards, quantile atoms, rejection, and proposal-support mismatch; additional regularity assumptions are stated where needed.

For a continuation law Q, let F_Q be the distribution function of r and define

$$R_{\alpha,Q} = \frac{1}{\alpha} \int_0^\alpha F_Q^{-1}(v)\, dv, \qquad F_Q^{-1}(v) = \inf\{z : F_Q(z) \geq v\}, \qquad 0 < \alpha \leq 1. \tag{29}$$

Here α specifies the target tail; β remains LRSI's proposal-tail parameter. The registered ON/OFF intervention must define the same function r on both supports. All statements concern population contributions conditional on this fixed context.

Lemma 1 (Allocation under a measure capacity)**.** *Let P be a probability measure, $C > 0$, $0 \leq m \leq C$, and $q(v) = F_P^{-1}(v)$ for $0 < v < 1$. Then*

$$\inf_{\substack{0 \leq \mu \leq CP \\ \mu(\Omega) = m}} \int (r + 1)\, d\mu = C \int_0^{m/C} (q(v) + 1)\, dv. \tag{30}$$

The infimum is attained by filling the lowest reward levels and taking a fractional amount of the boundary level when necessary. If only $\mu(\Omega) \geq m$ is required, the same expression is a lower bound for $\int (r + 1)\, d\mu$.

Proof. Set $t = m/C$. The cases $t = 0$ and $t = 1$ follow from the zero measure and the fact that $0 \leq \mu \leq CP$ with equal total mass forces $\mu = CP$, respectively. Assume $0 < t < 1$ and set $z = q(t)$. The definition of a generalized inverse and right continuity of the distribution function imply

$$p_- := P(r < z) = F_P(z-) \leq t \leq F_P(z) =: p_+. \tag{31}$$

$y < z$ implies $F_P(y) < t$, so taking $y \uparrow z$ gives the first inequality. For each k there is $y_k < z + 1/k$ with $F_P(y_k) \geq t$; monotonicity gives $F_P(z + 1/k) \geq t$, and right continuity gives the second inequality. Define

$$w_* = \mathbf{1}_{\{r<z\}} + \lambda \mathbf{1}_{\{r=z\}}, \qquad \lambda = \begin{cases} (t - p_-)/(p_+ - p_-), & p_+ > p_-, \\ 0, & p_+ = p_-. \end{cases} \tag{32}$$

When $p_+ = p_-$, Equation (31) gives $t = p_- = p_+$. Thus in every case $0 \leq w_* \leq 1$ and $\int w_* \, dP = t$. Any admissible μ has a Radon–Nikodym derivative Cw with $0 \leq w \leq 1$ and $\int w \, dP = t$. On $\{r < z\}$, $w - w_* \leq 0$; on $\{r > z\}$, $w - w_* \geq 0$. Consequently,

$$\begin{aligned} (r - z)(w - w_*) \geq 0 \quad & P\text{-almost surely}, \\ \int (r+1) w \, dP - \int (r+1) w_* \, dP &= \int (r - z)(w - w_*) \, dP + (z+1) \int (w - w_*) \, dP \\ &= \int (r - z)(w - w_*) \, dP \geq 0. \end{aligned} \tag{33}$$

This proves optimality, including tied rewards.

The optimizer's value follows from the generalized inverse, including at atoms. If U is uniform on $(0, 1)$, then for every real y, the generalized-inverse equivalence gives $\{v : q(v) \leq y\} = (0, F_P(y)]$ up to endpoints. Therefore $q(U)$ has the same law as r under P. The interval $(p_-, p_+]$ is mapped to z, up to null endpoints, whereas the portion below p_- has exactly the distribution of rewards strictly below z. It follows that

$$\begin{aligned} C \int_0^t (q(v) + 1) \, dv &= C \int_{\{r<z\}} (r+1) \, dP + C(t - p_-)(z+1) \\ &= C \int (r+1) w_* \, dP \\ &\leq C \int (r+1) w \, dP = \int (r+1) \, d\mu. \end{aligned} \tag{34}$$

Since Cw_*P is admissible, equality holds at the infimum. Finally, if an admissible measure has mass $M \geq m$, the case $m = 0$ follows from $r + 1 \geq 0$. For $m > 0$, the thinned measure $\widetilde{\mu} = (m/M)\mu$ satisfies

$$\begin{aligned} 0 \leq \widetilde{\mu} \leq \mu \leq CP, \qquad & \widetilde{\mu}(\Omega) = m, \\ \int (r+1) \, d\mu \geq \int (r+1) \, d\widetilde{\mu} &\geq C \int_0^{m/C} (q(v) + 1) \, dv. \end{aligned} \tag{35}$$

This establishes the final assertion. □

Lemma 2 (Submeasure and probability-envelope representations)**.** *For a probability measure Q and $0 < \alpha \leq 1$,*

$$\begin{aligned} R_{\alpha,Q} &= \frac{1}{\alpha} \min_{\substack{0 \leq \nu \leq Q \\ \nu(\Omega) = \alpha}} \int r \, d\nu \\ &= \min_{\substack{V \text{ probability} \\ V \ll Q,\ 0 \leq dV/dQ \leq 1/\alpha}} \int r \, dV. \end{aligned} \tag{36}$$

Moreover, $R_{\alpha,Q}$ is nondecreasing in α.

Proof. Apply Lemma 1 with $P = Q$, $C = 1$, and $m = \alpha$. Every feasible measure has the same mass, so subtracting that mass from the objective gives

$$\begin{aligned} \min_{\nu \leq Q,\, \nu(\Omega) = \alpha} \int r \, d\nu &= \min_{\nu \leq Q,\, \nu(\Omega) = \alpha} \int (r+1) \, d\nu - \alpha \\ &= \int_0^\alpha (F_Q^{-1}(v) + 1) \, dv - \alpha = \alpha R_{\alpha,Q}. \end{aligned} \tag{37}$$

The correspondence $V = \nu/\alpha$ is bijective between the two feasible sets in Equation (36), proving the second equality and attainment. For $0 < \alpha_1 \le \alpha_2 \le 1$, monotonicity of the quantile implies

$$\begin{aligned} R_{\alpha_2,Q} &= \frac{\alpha_1}{\alpha_2} R_{\alpha_1,Q} + \frac{1}{\alpha_2} \int_{\alpha_1}^{\alpha_2} F_Q^{-1}(v)\, dv \\ &\ge \frac{\alpha_1}{\alpha_2} R_{\alpha_1,Q} + \frac{\alpha_2 - \alpha_1}{\alpha_2} R_{\alpha_1,Q} = R_{\alpha_1,Q}. \end{aligned} \tag{38}$$

Indeed, every quantile above α_1 is at least the average of the quantiles below it. This also proves monotonicity at atoms. □

D.4 Clipping a target distribution against the proposal law

A density-ratio assumption alone excludes successors outside the proposal support. Write $Q = fP + Q_\perp$ for its Lebesgue decomposition, with $Q_\perp \perp P$. To admit such outcomes, for $C \ge 1$ define the excess mass

$$\varepsilon_C(Q\|P) = \sup_B \{Q(B) - CP(B)\} = Q_\perp(\Omega) + \int (f - C)_+\, dP. \tag{39}$$

The supremum is over measurable continuation events. This is the classical hockey-stick divergence, also denoted E_C; at $C = 1$ it equals total variation (Sason & Verdú, 2016). Our analysis combines this measure decomposition with the CVaR risk envelope (Rockafellar & Uryasev, 2002; Rockafellar et al., 2006).

Lemma 3 (Excess mass and maximal covered submeasure)**.** *Let P, Q be probability measures and $C \ge 1$. For the Lebesgue decomposition $Q = fP + Q_\perp$, set*

$$Q_0 = \min\{f, C\}P, \qquad E = (f - C)_+ P + Q_\perp, \qquad e_C = E(\Omega). \tag{40}$$

Then $Q = Q_0 + E$, $Q_0 \le CP$, $e_C = \varepsilon_C(Q\|P)$, and

$$1 - e_C = \max\{\mu(\Omega) : 0 \le \mu \le Q,\ \mu \le CP\}. \tag{41}$$

In particular, $\varepsilon_C = 0$ if and only if $Q \le CP$, and $\varepsilon_1(Q\|P) = \mathrm{TV}(P, Q)$.

Proof. Choose a measurable singular carrier Z with $P(Z) = 0$ and $Q_\perp(\Omega \setminus Z) = 0$. For any measurable B,

$$\begin{aligned} Q(B) - CP(B) &= \int_{B\setminus Z} (f - C)\, dP + Q_\perp(B \cap Z) \\ &\le \int_{B\setminus Z} (f - C)_+\, dP + Q_\perp(B \cap Z) \\ &\le \int (f - C)_+\, dP + Q_\perp(\Omega) = e_C. \end{aligned} \tag{42}$$

Take $B_* = Z \cup \{f > C\}$, choosing any version of f on Z. The nonpositive portion of the integral is then absent and all singular mass is included, giving

$$Q(B_*) - CP(B_*) = \int_{\{f>C\}} (f - C)\, dP + Q_\perp(Z) = e_C. \tag{43}$$

This proves the event and integral representations in Equation (39). The identity $f = \min\{f, C\} + (f - C)_+$ gives $Q = Q_0 + E$ and $Q_0(\Omega) = 1 - e_C$. For any $\mu \le CP$, write $\mu = gP$ with $0 \le g \le C$. If also $\mu \le Q$, restricting to $\Omega \setminus Z$ gives $g \le f$ almost surely. Therefore

$$\mu(\Omega) = \int g\, dP \le \int \min\{f, C\}\, dP = Q_0(\Omega) = 1 - e_C, \tag{44}$$

and equality is attained by Q_0. The zero-excess equivalence follows directly. Finally, $(Q-P)(\Omega) = 0$ implies $(Q - P)(B) = -(Q - P)(B^c)$, whence

$$\sup_B |Q(B) - P(B)| = \sup_B \{Q(B) - P(B)\} = \varepsilon_1(Q\|P). \tag{45}$$

□

D.5 SHARP TRANSFER UNDER SUCCESSOR DISTRIBUTION SHIFT

Theorem 3 (Sharp successor-shift transfer)**.** *Let $r : \Omega \to [-1, 1]$ be measurable and suppose $\varepsilon_C(Q\|P) \le \varepsilon$, where $C \ge 1$ and $0 \le \varepsilon \le 1$. For every $0 < \alpha \le 1$,*

$$R_{\alpha,Q} \ge \mathcal{B}_{\alpha,C,\varepsilon}(P) := -1 + \frac{C}{\alpha} \int_0^{(\alpha-\varepsilon)_+/C} \big(F_P^{-1}(v) + 1\big)\, dv. \tag{46}$$

For any prescribed baseline reward distribution on $[-1, 1]$, C, and ε, there exist a probability space and laws realizing that baseline distribution and $\varepsilon_C(Q\|P) = \varepsilon$ for which equality holds simultaneously for every α. Thus the bound is optimal given only that distribution, C, and ε over general probability spaces.

Proof of the transfer inequality. Use the decomposition in Lemma 3, and write $e_C \le \varepsilon$ for its actual excess mass. Fix any $\nu \le Q$ with $\nu(\Omega) = \alpha$. By Radon–Nikodym, $\nu = \theta Q$ for a measurable $0 \le \theta \le 1$. Define $\nu_0 = \theta Q_0$ and $\nu_E = \theta E$. The same multiplier is applied to both components, so

$$\begin{aligned} \nu = \nu_0 + \nu_E, \qquad & 0 \le \nu_0 \le Q_0 \le CP, \\ 0 \le \nu_E(\Omega) = \int \theta\, dE \le E(\Omega) = e_C \le \varepsilon, \\ \nu_0(\Omega) = \alpha - \nu_E(\Omega) \ge \max\{\alpha - e_C, 0\} \ge (\alpha - \varepsilon)_+. \end{aligned} \tag{47}$$

Put $m = (\alpha-\varepsilon)_+$. If $m = 0$, the claimed bound is simply $R_{\alpha,Q} \ge -1$. Otherwise the mass-thinning assertion of Lemma 1 applies to ν_0. Using $r + 1 \ge 0$ gives

$$\begin{aligned} \int r\, d\nu &= -\alpha + \int (r+1)\, d\nu \\ &= -\alpha + \int (r+1)\, d\nu_0 + \int (r+1)\, d\nu_E \\ &\ge -\alpha + \int (r+1)\, d\nu_0 \\ &\ge -\alpha + \inf_{\substack{0 \le \mu \le CP \\ \mu(\Omega) \ge m}} \int (r+1)\, d\mu \\ &= -\alpha + \min_{\substack{0 \le \mu \le CP \\ \mu(\Omega) = m}} \int (r+1)\, d\mu \\ &= -\alpha + C \int_0^{m/C} \big(F_P^{-1}(v) + 1\big)\, dv. \end{aligned} \tag{48}$$

The penultimate equality follows because every measure with larger mass can be thinned without increasing a nonnegative integral, and every measure of mass m belongs to the larger feasible set. Since the last expression is independent of ν, Lemma 2 yields

$$\begin{aligned} R_{\alpha,Q} &= \frac{1}{\alpha} \min_{\nu \le Q,\, \nu(\Omega) = \alpha} \int r\, d\nu \\ &\ge -1 + \frac{C}{\alpha} \int_0^{(\alpha-\varepsilon)_+/C} (F_P^{-1}(v) + 1)\, dv = \mathcal{B}_{\alpha,C,\varepsilon}(P). \end{aligned} \tag{49}$$

When $\varepsilon < \alpha$, expanding the constant inside the integral gives

$$\begin{aligned} \mathcal{B}_{\alpha,C,\varepsilon}(P) &= -1 + \frac{\alpha - \varepsilon}{\alpha} \big(R_{(\alpha-\varepsilon)/C,P} + 1\big) \\ &= \Big(1 - \frac{\varepsilon}{\alpha}\Big) R_{(\alpha-\varepsilon)/C,P} - \frac{\varepsilon}{\alpha}. \end{aligned} \tag{50}$$

All steps use integrals of bounded measurable functions and remain valid at reward atoms. □

Proof of simultaneous sharpness. Let F be any prescribed probability distribution on $[-1,1]$ and let $q = F^{-1}$ on $(0,1)$. Construct $\Omega = (0,1) \sqcup \{\star\}$ with its natural sigma algebra, let P be Lebesgue probability measure on $(0,1)$, and set

$$r(u) = q(u) \quad (0 < u < 1), \qquad r(\star) = -1, \qquad t = \frac{1-\varepsilon}{C}, \qquad Q = C \operatorname{Leb}|_{(0,t)} + \varepsilon\delta_\star. \tag{51}$$

The inverse-transform identity proved in Lemma 1 shows that the reward distribution under P is exactly F, including any atoms. Moreover,

$$Q(\Omega) = Ct + \varepsilon = 1, \qquad \varepsilon_C(Q\|P) = \varepsilon + \int (C\mathbf{1}_{(0,t)} - C)_+ \, dP = \varepsilon. \tag{52}$$

For $0 < \alpha \le \varepsilon$, choose the submeasure $\nu = \alpha\delta_\star$. It is feasible in Lemma 2, so $R_{\alpha,Q} \le -1$; boundedness gives equality, matching Equation (46). For $\alpha > \varepsilon$, set $t_\alpha = (\alpha - \varepsilon)/C \le t$ and choose $\nu_\alpha = \varepsilon\delta_\star + C \operatorname{Leb}|_{(0,t_\alpha)}$. Then $\nu_\alpha \le Q$ and $\nu_\alpha(\Omega) = \alpha$. Applying the already proved lower bound and this explicit feasible measure gives the two-sided chain

$$\begin{aligned} -1 + \frac{C}{\alpha}\int_0^{t_\alpha} (q(v)+1)\, dv \le R_{\alpha,Q} &\\ &\le \frac{1}{\alpha}\int r \, d\nu_\alpha \\ &= \frac{-\varepsilon + C\int_0^{t_\alpha} q(v)\, dv}{\alpha} \\ &= -1 + \frac{C}{\alpha}\int_0^{t_\alpha} (q(v)+1)\, dv. \end{aligned} \tag{53}$$

Thus equality holds for all α with a single pair (P, Q). The constructions also cover $\varepsilon = 0$ and $\varepsilon = 1$ through empty-interval conventions. The extremizer is constructed on an enlarged probability space, establishing sharpness across spaces with the specified baseline reward distribution. □

If $\varepsilon < \alpha$, the right side equals $(1 - \varepsilon/\alpha)R_{(\alpha-\varepsilon)/C,P} - \varepsilon/\alpha$; if $\varepsilon \ge \alpha$, it equals -1, without invoking a zero-level CVaR. The bound therefore distinguishes concentration within covered successors from probability mass that can enter the worst tail without proposal coverage.

D.6 Density, single-score, and total-variation consequences

Corollary 1 (Density domination and total variation)**.** *If $Q \le CP$, then $R_{\alpha,Q} \ge R_{\alpha/C,P}$. More generally, for $\beta = \alpha/C$,*

$$R_{\alpha,Q} \ge \max\left\{-1,\ R_{\beta,P} - \frac{2\varepsilon_C(Q\|P)}{\alpha}\right\}. \tag{54}$$

For $d = \mathrm{TV}(P,Q)$, the uniform same-tail modulus is

$$|R_{\alpha,Q} - R_{\alpha,P}| \le 2\min\{d/\alpha, 1\}, \tag{55}$$

and this modulus is attained for every $d \in [0,1]$ and $\alpha \in (0,1]$ by a two-point construction.

Proof of Corollary 1. If $Q \le CP$, Lemma 3 gives $\varepsilon_C = 0$. Theorem 3 then implies

$$R_{\alpha,Q} \ge -1 + \frac{C}{\alpha}\int_0^{\alpha/C} (F_P^{-1}(v) + 1)\, dv = R_{\alpha/C,P}. \tag{56}$$

To obtain a bound from a single baseline score, put $e = \varepsilon_C(Q\|P)$ and $\beta = \alpha/C$. When $e < \alpha$, Theorem 3 and $0 \le F_P^{-1} + 1 \le 2$ yield

$$\begin{aligned} R_{\alpha,Q} &\ge -1 + \frac{C}{\alpha}\int_0^{\beta - e/C} (F_P^{-1}(v)+1)\, dv \\ &= R_{\beta,P} - \frac{C}{\alpha}\int_{\beta-e/C}^{\beta} (F_P^{-1}(v)+1)\, dv \\ &\ge R_{\beta,P} - \frac{C}{\alpha}\cdot\frac{e}{C}\cdot 2 = R_{\beta,P} - \frac{2e}{\alpha}. \end{aligned} \tag{57}$$

If $e \geq \alpha$, then $R_{\beta,P} - 2e/\alpha \leq 1 - 2 = -1$, so the bound $R_{\alpha,Q} \geq -1$ proves the same assertion. Combining the range bound with both cases gives Equation (54).

For the TV statement, $C = 1$ and $e = d$. In the case $d < \alpha$, applying Equation (57) in both directions gives the more informative pair

$$\begin{aligned} R_{\alpha,P} - R_{\alpha,Q} &\leq \frac{1}{\alpha} \int_{\alpha-d}^{\alpha} (F_P^{-1}(v) + 1)\, dv \leq \frac{2d}{\alpha}, \\ R_{\alpha,Q} - R_{\alpha,P} &\leq \frac{1}{\alpha} \int_{\alpha-d}^{\alpha} (F_Q^{-1}(v) + 1)\, dv \leq \frac{2d}{\alpha}. \end{aligned} \tag{58}$$

For $d \geq \alpha$, both scores belong to $[-1, 1]$, so their difference is at most 2. This proves Equation (55) for all d. For exact sharpness, take a two-point reward space $\{-1, +1\}$ and laws $P = \delta_{+1}$ and $Q = d\delta_{-1} + (1 - d)\delta_{+1}$. Then $\mathrm{TV}(P, Q) = d$ and

$$R_{\alpha,P} = 1, \qquad R_{\alpha,Q} = \begin{cases} -1, & \alpha \leq d, \\ 1 - 2d/\alpha, & \alpha > d, \end{cases} \qquad |R_{\alpha,Q} - R_{\alpha,P}| = 2\min\{d/\alpha, 1\}. \tag{59}$$

The coefficient and truncation are therefore both necessary in a uniform bound. The full quantile profile in Theorem 3 can be strictly more informative for a specified baseline law; the range-only modulus is already optimal without that additional information. □

D.7 Proof of the main-text guarantee

Proof of Theorem 1. Take $\alpha = 1$, $C = 1/\beta$, and $e = \varepsilon_\beta = \varepsilon_{1/\beta}(Q\|P)$ in Theorem 3. Since $0 < \beta \leq 1$, the density cap satisfies $C \geq 1$; since P, Q are probability laws, $e \in [0, 1]$. The quantile representation of expectation and $0 \leq F_P^{-1}(v) + 1 \leq 2$ give the complete chain

$$\begin{aligned} \mathbb{E}_Q r = R_{1,Q} &\geq -1 + \frac{1}{\beta} \int_0^{\beta(1-e)} (F_P^{-1}(v) + 1)\, dv \\ &= R_{\beta,P} - \frac{1}{\beta} \int_{\beta(1-e)}^{\beta} (F_P^{-1}(v) + 1)\, dv \\ &\geq R_{\beta,P} - \frac{1}{\beta} \cdot \beta e \cdot 2 = R_{\beta,P} - 2\varepsilon_\beta. \end{aligned} \tag{60}$$

The same chain includes $e = 0$ and $e = 1$, because an integral over an empty interval is zero. Combining it with $r \geq -1$ gives Equation (8). If $Q \leq P/\beta$, every event discrepancy in the definition of ε_β is nonpositive, while the empty event has discrepancy zero; hence $\varepsilon_\beta = 0$. Thus $R_{\beta,P} \geq 0$ implies $\mathbb{E}_Q r \geq R_{\beta,P} \geq 0$. □

Thus the population gate tolerates reweighting up to $1/\beta$; mass beyond this cap incurs an explicit penalty. Appendix D.5 gives the sharp quantile bound, while Appendix D.8 characterizes the concentration induced by finite-pool selection and its tight zero-gate threshold. This guarantee concerns population retained contribution; applying it to empirical acceptance requires estimation control (Appendix D.2).

D.8 Finite-pool selection: exact transport and its sharp threshold

Selection convention. All probabilities in this subsection condition on the fixed current harness, the fixed edit whose contribution is being measured, and the generation context appearing in Theorem 4. The raw proposals $X_1, \ldots, X_N$ are independent with common law P_0. Write S for the measurable set of nonempty, structurally valid proposals, $s = P_0(S) > 0$, and $P(B) = P_0(B \cap S)/s$. A selector may inspect the entire pool and arbitrary additional randomness, apply any further screening, and remove duplicates, but it either returns an observed element of S or returns no update. Let A be its acceptance event, let $a = \Pr(A) > 0$, and let Q_+ be the conditional law of the returned proposal on A. The retained-contribution function $r : S \to [-1, 1]$ must be defined on the entire relevant support by the intervention contract. This support condition applies to every relevant successor, including those outside the finite probe sample. Let $F_P(t) = P(r \leq t)$ and

$F_P^{-1}(u) = \inf\{t \in [-1,1] : F_P(t) \geq u\}$ for $u \in (0,1)$; endpoint values of the inverse do not affect any integral below. Define the pool capacity and its inverse on the finite part of their domains by

$$H(u) = 1-(1-su)^N, \qquad H^{-1}(v) = \frac{1-(1-v)^{1/N}}{s}, \qquad 0 \leq u \leq 1, \quad 0 \leq v \leq H(1). \tag{61}$$

The acceptance convention implies $0 < a \leq H(1)$.

Theorem 4 (Sharp transfer through a finite proposal pool)**.** *Under the preceding conditions, $Q_+ \leq (Ns/a)P$ and, for every $0 < \alpha \leq 1$,*

$$\begin{aligned} R_{\alpha,Q_+} &\geq \frac{1}{\alpha a}\int_0^{\alpha a} F_P^{-1}\left(\frac{1-(1-v)^{1/N}}{s}\right) dv \\ &\geq R_{\alpha a/(Ns),P}. \end{aligned} \tag{62}$$

For any baseline reward distribution and feasible N, s, a, a rule that selects a minimum-contribution valid proposal and accepts only its lowest a probability mass, randomizing at a boundary atom, attains the first inequality simultaneously for every α.

The inverse-pool expression retains information discarded by the density cap Ns/a. The assumption $P = P_0(\cdot \mid S)$ holds the generator and its input evidence fixed.

Proof of Theorem 4. We first prove the eventwise capacity inequality and then integrate it to obtain the exact bound. For every measurable $B \subseteq S$, let $\mu(B) = \Pr(A,\, X_* \in B) = aQ_+(B)$, where X_* denotes the returned proposal only on A. Since returning an element of B requires that the raw pool contain an element of B, independence and the union bound give the chain

$$\begin{aligned} \mu(B) &= \Pr(A,\, X_* \in B) \\ &\leq \Pr\left(\bigcup_{i=1}^N \{X_i \in B\}\right) \\ &= 1 - \prod_{i=1}^N \Pr(X_i \notin B) \\ &= 1 - \big(1 - P_0(B)\big)^N \\ &= 1 - \big(1 - sP(B)\big)^N \\ &= H(P(B)) \\ &\leq NsP(B). \end{aligned} \tag{63}$$

Consequently $\mu \ll P$, and the Radon–Nikodym theorem and the last inequality imply

$$0 \leq \frac{d\mu}{dP} \leq Ns \quad P\text{-almost surely}, \qquad 0 \leq \frac{dQ_+}{dP} \leq \frac{Ns}{a} \quad P\text{-almost surely}. \tag{64}$$

If $d\mu/dP > Ns$ on a set of positive P-measure, then some set $B_k = \{d\mu/dP \geq Ns + 1/k\}$ has positive measure, and integrating over B_k contradicts $\mu(B_k) \leq NsP(B_k)$. The density conclusion follows from the inequality over all measurable sets.

Fix a target tail fraction $\alpha \in (0,1]$ and write $z = \alpha a$. By Lemma 2, a feasible lower-tail submeasure is any $\nu \leq Q_+$ with total mass $\nu(S) = \alpha$. For such a ν, put $m = a\nu$. Then $m(S) = z$ and $m \leq \mu$, so Equation (63) gives, for every $t \in [-1,1]$,

$$\begin{aligned} m(r \leq t) &\leq \min\{m(S), \mu(r \leq t)\} \\ &\leq \min\{z, H(F_P(t))\}. \end{aligned} \tag{65}$$

For any finite measure m of mass z and any $r \in [-1,1]$, Tonelli's theorem applied to $1 - r = \int_{-1}^1 \mathbf{1}\{r \leq t\}\, dt$ yields the signed-reward layer-cake identity. Combining that identity with Equa-

tion (65) gives

$$\begin{aligned}\int_S r\,dm &= z - \int_{-1}^{1} m(r \le t)\,dt \\ &\ge z - \int_{-1}^{1} \min\{z, H(F_P(t))\}\,dt \\ &= \int_0^z F_P^{-1}\big(H^{-1}(v)\big)\,dv.\end{aligned} \tag{66}$$

Here the final equality remains exact when r has atoms. Indeed, for Lebesgue-almost every $v \in (0, z)$, the generalized-inverse identity $F_P^{-1}(u) \le t$ if and only if $u \le F_P(t)$ gives

$$\begin{aligned}\int_0^z \mathbf{1}\big\{F_P^{-1}\big(H^{-1}(v)\big) \le t\big\}\,dv &= \int_0^z \mathbf{1}\{H^{-1}(v) \le F_P(t)\}\,dv \\ &= \int_0^z \mathbf{1}\{v \le H(F_P(t))\}\,dv \\ &= \min\{z, H(F_P(t))\}.\end{aligned} \tag{67}$$

Integrating Equation (67) over $t \in [-1, 1]$ proves the equality in Equation (66) directly, including for discrete contribution distributions. Taking the infimum over all feasible ν therefore proves the first, exact finite-pool bound:

$$\begin{aligned}R_{\alpha,Q_+}(r) &= \inf_{\substack{0\le\nu\le Q_+ \\ \nu(S)=\alpha}} \frac{1}{\alpha}\int_S r\,d\nu \\ &= \inf_{\substack{0\le\nu\le Q_+ \\ \nu(S)=\alpha}} \frac{1}{z}\int_S r\,d(a\nu) \\ &\ge \frac{1}{z}\int_0^z F_P^{-1}\big(H^{-1}(v)\big)\,dv \\ &= \frac{1}{\alpha a}\int_0^{\alpha a} F_P^{-1}\left(\frac{1-(1-v)^{1/N}}{s}\right)\,dv.\end{aligned} \tag{68}$$

To obtain the CVaR comparison with a single tail parameter, one can either apply the capped-submeasure characterization to $m \le NsP$ or compare quantiles pointwise. The latter makes the relaxation explicit. Since $H(u) \le Nsu$, inversion gives $H^{-1}(v) \ge v/(Ns)$ for $0 \le v \le H(1)$. Because F_P^{-1} is nondecreasing, a change of variables yields

$$\begin{aligned}\frac{1}{z}\int_0^z F_P^{-1}\big(H^{-1}(v)\big)\,dv &\ge \frac{1}{z}\int_0^z F_P^{-1}\Big(\frac{v}{Ns}\Big)\,dv \\ &= \frac{Ns}{z}\int_0^{z/(Ns)} F_P^{-1}(u)\,du \\ &= R_{z/(Ns),P}(r) \\ &= R_{\alpha a/(Ns),P}(r).\end{aligned} \tag{69}$$

The tail parameter is admissible because $0 < z \le a \le H(1) \le Ns$. Equation (69) also identifies exactly where information is lost: the finite-pool event capacity H is replaced by its tangent $u \mapsto Nsu$ at zero.

We next establish sharpness of the finite-pool bound, simultaneously for every $\alpha \in (0, 1]$ and for arbitrary atoms of the reward law. Enlarge the selector's probability space with independent uniform variables $V_1, \dots, V_N$ on $(0, 1)$. For a valid proposal define its randomized rank, and for an invalid proposal assign an infinite rank:

$$U_i = \begin{cases} F_P(r(X_i)-) + V_i\big[F_P(r(X_i)) - F_P(r(X_i)-)\big], & X_i \in S, \\ +\infty, & X_i \notin S. \end{cases} \tag{70}$$

Conditional on $X_i \in S$, the randomized probability transform makes U_i uniform on $(0, 1)$ and gives $r(X_i) = F_P^{-1}(U_i)$ almost surely. The transform also covers atoms: if $x = F_P^{-1}(u)$, the mass strictly

below x contributes $F_P(x-)$, and the required fraction of the atom at x contributes $u - F_P(x-)$, giving total probability u; when the atom has zero mass the same conclusion follows by the defining generalized-inverse inequalities and continuity of the distribution function at x. Independence is preserved because the auxiliary variables are independent across proposals.

Let $Z = \min_{1\le i\le N} U_i$, let $u_a = H^{-1}(a)$, and accept precisely when $Z \le u_a$. On that event select an index attaining the minimum rank, breaking the null-probability rank ties by proposal order. If all proposals are invalid, $Z = +\infty$ and the selector abstains. For $u \in [0,1]$ the rank-minimum distribution and the acceptance probability satisfy

$$\begin{aligned}\Pr(Z \le u) &= 1 - \prod_{i=1}^{N} \Pr(U_i > u) \\ &= 1 - (1 - su)^N \\ &= H(u), \qquad \Pr(Z \le u_a) = H(u_a) = a.\end{aligned} \tag{71}$$

Thus $H(Z)/a$ is uniform on $(0,1)$ conditional on acceptance, while the selected reward is $F_P^{-1}(Z)$. Its quantile function is therefore $u \mapsto F_P^{-1}(H^{-1}(au))$ almost everywhere, since this map is nondecreasing. Equivalently, the selected distribution saturates every contribution-sublevel capacity:

$$\begin{aligned}Q_+(r \le t) &= \frac{\Pr(Z \le u_a,\ F_P^{-1}(Z) \le t)}{a} \\ &= \frac{\Pr(Z \le \min\{u_a, F_P(t)\})}{a} \\ &= \frac{H(\min\{u_a, F_P(t)\})}{a} \\ &= \min\left\{1, \frac{H(F_P(t))}{a}\right\}.\end{aligned} \tag{72}$$

The quantile formula then yields equality in Equation (68) for every α under this one selector. It chooses the smallest observed contribution and accepts the lowest a probability mass of the resulting pool minima; randomized ranks implement the necessary randomization inside any tied contribution atom. This auxiliary selector establishes the universal bound's sharpness.

Finally, Ns/a is the smallest uniform density constant. Choose a valid-law event B of mass $p \in (0, u_a)$, give B a strictly smaller reward than its complement, and use the preceding extremal selector. The event B is selected whenever the pool contains it, so

$$\frac{Q_+(B)}{P(B)} = \frac{H(p)}{ap} = \frac{1-(1-sp)^N}{ap} \xrightarrow[p\downarrow 0]{} \frac{Ns}{a}. \tag{73}$$

These examples can be realized by a two-point valid proposal law with an additional invalid point of raw mass $1 - s$. For $N > 1$, sharpness of the density constant is a limiting statement; the nonlinear finite-pool bound retains the strictly stronger eventwise information at positive event masses. □

Proposition 3 (Sharp zero-gate condition and a no-update contribution)**.** *Under Theorem 4, let $\beta \in (0,1]$. If $R_{\beta,P}(r) \ge 0$ and $\beta \le a/(Ns)$, then the accepted successor has nonnegative expected retained contribution. More precisely, with $\gamma = a/(Ns)$,*

$$\mathbb{E}_{Q_+}[r] \ge \frac{1}{a}\int_0^a F_P^{-1}\left(\frac{1-(1-v)^{1/N}}{s}\right) dv \ge R_{\gamma,P}(r) \ge R_{\beta,P}(r) \ge 0. \tag{74}$$

If the edit has fixed immediate population contribution $g_0 \ge 0$, let $\widetilde{r}$ equal $r(X_*)$ when an update is accepted and g_0 when no update is accepted. Then

$$\mathbb{E}[\widetilde{r}] \ge (1-a)g_0 + aR_{\gamma,P}(r) \ge (1-a)g_0 + aR_{\beta,P}(r) \ge (1-a)g_0 \ge 0. \tag{75}$$

Conversely, for every admissible N, s, a and every $\beta > a/(Ns)$, there exist a two-point valid proposal law, a reward $r \in [-1,1]$, and an admissible selector with acceptance probability exactly a for which $R_{\beta,P}(r) = 0$ but $\mathbb{E}_{Q_+}[r] < 0$. Thus $\beta \le a/(Ns)$ is the sharp universal condition for certification based only on the zero population gate.

Proof. Theorem 4 at $\alpha = 1$ proves the first two inequalities of Equation (74). For the third, the capped-density feasible sets are nested when $\beta \le \gamma$:

$$\begin{aligned} R_{\gamma,P}(r) &= \inf_{\substack{0\le w\le 1/\gamma \\ \mathbb{E}_P[w]=1}} \mathbb{E}_P[wr] \\ &\ge \inf_{\substack{0\le w\le 1/\beta \\ \mathbb{E}_P[w]=1}} \mathbb{E}_P[wr] \\ &= R_{\beta,P}(r) \\ &\ge 0. \end{aligned} \tag{76}$$

Conditioning on acceptance and using the fixed no-update contribution gives the complete unconditional calculation

$$\begin{aligned} \mathbb{E}[\widetilde{r}] &= \Pr(A^c)g_0 + \Pr(A)\mathbb{E}[r(X_*) \mid A] \\ &= (1-a)g_0 + a\mathbb{E}_{Q_+}[r] \\ &\ge (1-a)g_0 + \int_0^a F_P^{-1}\big(H^{-1}(v)\big)\, dv \\ &\ge (1-a)g_0 + aR_{\gamma,P}(r) \\ &\ge (1-a)g_0 + aR_{\beta,P}(r) \\ &\ge (1-a)g_0 \\ &\ge 0. \end{aligned} \tag{77}$$

Here $g_0 \ge 0$ is an assumption on the immediate population contribution; applying it after development-set screening requires estimation control.

For the converse, suppose $\beta > a/(Ns)$. Since $H'(0) = Ns > a/\beta$ and $u_a = H^{-1}(a) > 0$, there exists

$$0 < p < \min\{\beta/2, u_a\} \quad \text{such that} \quad \frac{H(p)}{p} > \frac{a}{\beta}. \tag{78}$$

Take valid proposals b and c with $P(\{b\}) = p$ and $P(\{c\}) = 1-p$, and set

$$r(b) = -1, \qquad r(c) = \frac{p}{\beta - p}. \tag{79}$$

The restriction $p < \beta/2$ guarantees $0 < r(c) < 1$. The raw law gives masses sp, $s(1-p)$, and $1-s$ to b, c, and an invalid symbol, respectively. The lower β fraction consists of all bad-proposal mass and $\beta - p$ good-proposal mass, and consequently

$$\begin{aligned} R_{\beta,P}(r) &= \frac{1}{\beta}\left[-p + (\beta - p)\frac{p}{\beta - p}\right] \\ &= 0. \end{aligned} \tag{80}$$

Use the extremal selector from the theorem proof. Because $p < u_a$, every pool containing b is accepted and returns b. The remaining accepted pools return c, with randomization among their identical rewards arranged by Equation (70). Equivalently, accept every pool containing b, and accept each pool containing at least one c and no b with the independent probability $(a - H(p))/(H(1) - H(p))$, which belongs to $[0,1]$. Either construction has acceptance probability exactly a and conditional bad-proposal mass $H(p)/a$. Its expected retained contribution satisfies the strict chain

$$\begin{aligned} \mathbb{E}_{Q_+}[r] &= -\frac{H(p)}{a} + \left(1 - \frac{H(p)}{a}\right)\frac{p}{\beta - p} \\ &= \frac{p - \beta H(p)/a}{\beta - p} \\ &< \frac{p - \beta\,(ap/\beta)/a}{\beta - p} \\ &= 0. \end{aligned} \tag{81}$$

This proves necessity of the stated threshold for a universal implication from $R_{\beta,P}(r) \ge 0$ to nonnegative conditional expected contribution. The same construction with $g_0 = 0$ also has negative unconditional contribution. □

If abstention is represented by $r(\varnothing) = g_{0,D}(e)$, then the complete next-update law satisfies

$$\mathbb{E}_{q_{\mathrm{sel}}} r \geq (1-a) g_{0,D}(e) + a R_{a/(Ns),P}. \tag{82}$$

This is the unconditional consequence proved in Proposition 3.

Generation context. The finite-pool theorem compares the probe law with a pool drawn from the same raw law P_0. Refreshing generation tasks or traces can change the raw law even with a frozen proposal generator. Such a change requires an explicit comparison between the actual next-round raw law and the probe raw law, or an application of the distribution-shift theorem to the resulting selected law. Equation (63) remains valid under further score-based screening, duplicate removal, and additional lookahead, since these operations only remove potential outputs from the observed pool. The theorem measures retained contribution under a fixed intervention and task distribution, separately from total successor utility and multi-round trajectory gain.

D.9 Composing selection with changed generation evidence

Proposition 4 (Composition of amplification and residual shift)**.** *For probability measures $P, \widetilde{P}, Q$ and constants $C, \Gamma \geq 1$,*

$$\varepsilon_{C\Gamma}(Q\|P) \leq \min\{1,\ \varepsilon_C(Q\|\widetilde{P}) + C\varepsilon_\Gamma(\widetilde{P}\|P)\}. \tag{83}$$

In particular, if the actual next-round valid-proposal law is $\widetilde{P}$ and selection gives $Q_+ \leq C\widetilde{P}$, then $\varepsilon_{C\Gamma}(Q_+\|P) \leq \min\{1, C\varepsilon_\Gamma(\widetilde{P}\|P)\}$.

Proof. For every measurable B, add and subtract $C\widetilde{P}(B)$, keeping the direction of each one-sided discrepancy:

$$\begin{aligned} Q(B) - C\Gamma P(B) &= Q(B) - C\widetilde{P}(B) + C\big(\widetilde{P}(B) - \Gamma P(B)\big) \\ &\leq \sup_A\{Q(A) - C\widetilde{P}(A)\} + C\sup_A\{\widetilde{P}(A) - \Gamma P(A)\} \\ &= \varepsilon_C(Q\|\widetilde{P}) + C\varepsilon_\Gamma(\widetilde{P}\|P). \end{aligned} \tag{84}$$

Taking the supremum over B proves the additive bound. The left side is at most 1 by its event representation, giving the truncation. If $Q_+ \leq C\widetilde{P}$, the first excess term vanishes by Lemma 3. □

This composition separates selection-induced concentration from a changed raw proposal context. With $\Gamma = 1$, a known TV discrepancy d between valid-proposal laws gives an excess budget at most Cd after selection; applying Theorem 3 then yields a population contribution certificate. Different harnesses and generation traces can yield positive d even with a fixed model, prompt template, and temperature.