REPEATABLE BUT NOT REPLICABLE: EVALUATOR STATE AS A HIDDEN FACET IN SELF-EVOLVING AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Self-evolving agents improve by keeping new versions of their prompts or code that score higher on test tasks. Exact reruns often check such comparisons, but evaluators also write incidental strings, such as run directories and identifiers, into model inputs, and a rerun repeats them, hiding their effect. We treat these strings, the evaluator state, as a measurement facet: sharing a state cancels only the changes two versions share. In an unmodified SQL-writing agent, a new equal-length identifier changes 48 of 240 first queries but only 4 grades. Decisions on 100 tasks hold within a study, but on 10 tasks two evaluations of the same pair disagree on a median of 13 percent of decisions, and sharing a state did not measurably help. When a GEPA prompt-optimization loop shows the model its run directory, a rerun repeats every step with a database clock fixed, but other directories keep different prompts. On 12 validation tasks, the added lines lower the bar a new version must beat, and 14 of 18 loops keep a version worse on held-out tasks. Without the lines, these losses persist but mostly vanish at 100 validation tasks. Evaluations should rerun comparisons under fresh evaluator states.

1 INTRODUCTION

Self-evolving agents improve themselves by proposing changes to their prompts, tools, memory or code and keeping the candidate versions that score better (Yuksekgonul et al., 2025; Hu et al., 2025; Zhang et al., 2026; Agrawal et al., 2026; Robeyns et al., 2025). A loop derives a new version, the *child*, from a current one, the *parent*, and keeps the child only if it scores higher. These comparisons are often small. GEPA, a widely used prompt optimizer, admits a child after it beats its parent on three training tasks and then selects among versions on a validation set (Agrawal et al., 2026). To make such comparisons repeatable, systems decode deterministically, cache model calls and rerun. DSPy, a framework for language-model pipelines, caches model calls by default (Khattab et al., 2024; DSPy, 2026), and GEPA's authors remove sampling noise "by operating under a cache" (Agrawal et al., 2026). An exact rerun, however, shows only that a result holds under the inputs it repeats.

The evaluator is not only a measuring instrument. It also writes part of the model's input. A harness, the program that runs an agent on a task, may tell the model its working directory, a folder such as `.../pi_3f9a2c71.../` named by a hash, and how long each tool call took. These strings are irrelevant to the task, but the model reads them, and its answers can change. Folder names and identifiers differ across runs in deployment, yet an exact rerun repeats them, so agreement between reruns says nothing about their effect. We call these strings the *evaluator state*. An exact rerun can therefore hide that an incidental detail, such as a folder name, changes which version a self-evolving agent keeps.

We study an agent that answers questions about databases by writing SQL queries. A folder name reaches what a loop keeps through four links: the model's *response*, the task's *grade*, the *retention decision* and the kept version's *held-out accuracy* on tasks the loop never used. A changed response need not change a grade, and a changed grade need not change a decision. Figure 2 shows how grades connect to a keep/reject decision. In one archived loop, the same child answers 4 of 12 validation tasks in one directory and 5 in another. Its parent, the loop's initial prompt, answers 3 and 5. The child's score rises, yet it loses its strict advantage because the parent's score rises more. The initial

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.

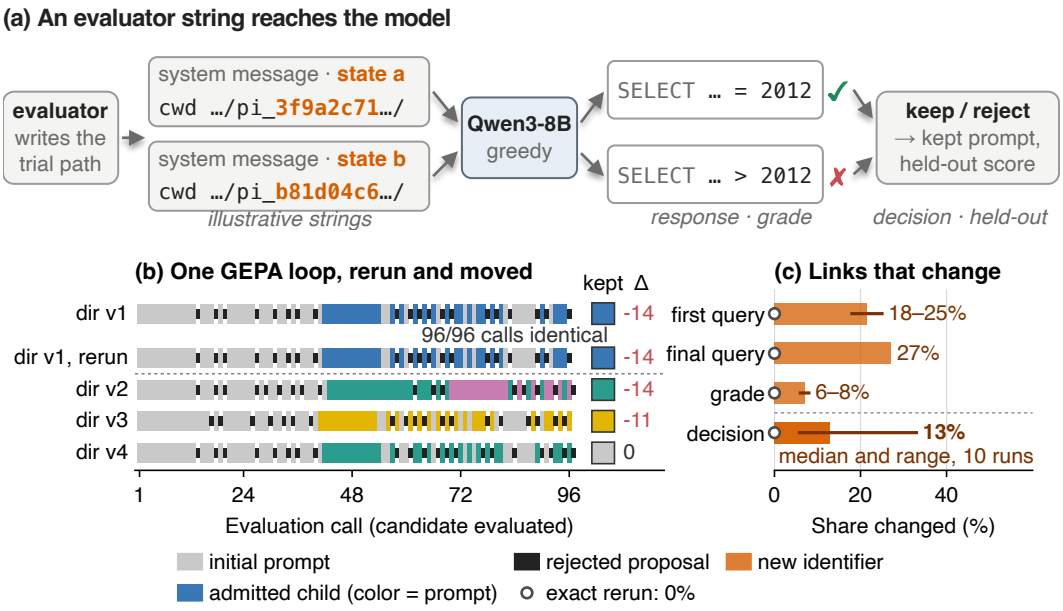


Figure 1: **Exact reruns repeat the comparison; fresh evaluator states change it.** (a) The evaluator writes a hashed folder path into the system message (strings illustrative). (b) Which version one GEPA loop evaluates at each evaluation call in directory v1, a rerun of v1 and directories v2–v4 (12 validation tasks, greedy decoding, clock fixed); squares: the kept prompt; Δ: its held-out change over the initial prompt, in accuracy points. (c) Share that a new identifier changes, of queries and grades on 100 test tasks for one archived parent–child pair (bars: mean of parent and child; whiskers: range) and of 10-task keep/reject decisions (median and range over ten runs of five pairs).

prompt's score is thus the *bar* a child must beat, and in our loops, directories that lower the bar mostly keep worse children.

Figure 1 previews what changes when only the evaluator state changes. We first fix a database clock that SQL queries can read. An exact rerun of a GEPA loop whose system message shows its directory then repeats all 96 of its evaluation calls and keeps the same prompt. Moving the loop to another directory changes the versions it evaluates from the first calls and changes the prompt it keeps (Figure 1b). Figure 1c traces a new identifier along the four links.

In this paper, we treat evaluator state as a *facet*, a term from generalizability theory for a condition of measurement, like the rater in human grading, over which a score may need to generalize (Cronbach et al., 1972; Brennan, 2001). *Repeatability* is agreement at a fixed state. *Replicability across states* is agreement after fresh states are drawn. Giving both versions the same state cancels only the changes they share. If a folder name lowers both scores, a shared name cancels that shift; if it changes only one version, sharing does not help. Unlike studies of prompt sensitivity and noisy selection (Salinas & Morstatter, 2024; Sclar et al., 2024; Polo et al., 2024; Smith & Winkler, 2006), we study context that the evaluator itself writes, which exact reruns preserve and compared versions need not share. Our main contributions are as follows:

- **Evaluator state reaches the model.** We isolate two routes in an unmodified agent harness, an identifier in a folder path and a measured tool latency, and find such strings in most audited harnesses (Section 2).
- **Fresh states split small decisions.** For archived versions, decisions on 100 tasks hold within each study, but many 10-task decisions flip, and sharing a state did not measurably help (Sections 3–4).
- **The directory decides what a loop keeps.** Once a run directory is shown to the model, the directory alone changes which prompt GEPA and OPRO (Yang et al., 2024) loops keep, in controls registered in advance. The added lines also lower the bar, and most small loops keep a worse version (Section 5).

- **Stable decisions need many tasks.** We estimate the score gap a stable decision needs and recommend rerunning comparisons under fresh evaluator states, not exactly (Sections 5.3 and 7).

2 Evaluator State in Model Inputs

We audit an unmodified agent harness, the Pi runner in the HarnessLens code at a pinned commit, on BIRD, a benchmark of questions over databases (Xu et al., 2026; Zechner, 2026; Li et al., 2023). The agent answers in SQL, and a grader compares the result of its query with that of a reference query. Table 1 summarizes the interventions.

Identifier channel. For each trial, the harness names the working folder with a 20-character hash of its full path. It shows this folder and the path of an `AGENTS.md` instructions file in the system message. To test the effect, we replace only this hash with another string of the same token length, chosen in advance, while the agent keeps working in the same folder. Instructions, tasks, databases, tools, weights, decoding, budgets and grader stay fixed. On 60 training-split tasks with two Qwen3 sizes and two context configurations (Yang et al., 2025; Peng et al., 2024), the identifier changes 48 of 240 first queries and 4 grades. All 240 exact reruns match. With Mistral-7B, it changes 43 of 100 queries and 5 grades, with no favored direction (Appendix B).

Table 1: **Evaluator strings change responses far more often than grades.** Tasks affected, as changed/eligible. Rerun: the same request at the same state, compared on responses (rows 1–2) or full trajectories (rows 3–4). New identifier: a replacement string of equal token length. Row 3 counts the 30 tasks whose reruns match; row 4 drops one clock-reading task.

	Rerun	New identifier changes		
Setting (unit)	differs	first query	final query	grade
Qwen3, first turn (60 tasks × 4 configurations)	0/240	48/240	–	4/240
Mistral-7B, one SQL fence (tasks)	0/100	43/100	43/100	5/100
Qwen3-8B episodes, latency visible (tasks)	30/60	7/30	10/30	0/30
Qwen3-8B episodes, latency removed (tasks)	0/59	14/59	18/59	1/59

Latency channel. After each SQL call the evaluator reports the elapsed time, `elapsed_s`, in the tool result. In 60 complete Qwen3-8B episodes, exact reruns repeat every grade but only 30 trajectories, the full sequences of actions and observations. The first mismatch is always this field. Without it, 59 of 60 trajectories repeat; the last differs because its SQL reads the database clock. With the field removed, a new identifier still changes 18 of these 59 final queries.

From responses to decisions. Grades change far less often than queries, but decisions compare few tasks. We scored two byte-identical copies of one candidate under their own identifiers on five blocks of 11 or 12 tasks. Accuracy alone prefers one copy in one block. A common tie-break, fewer calls and tokens, prefers one copy in all five. Every such preference is spurious, because the two copies are identical.

Exposure in other harnesses. Eight of nine audited agent harnesses show a working directory to the model; three of these paths never change, such as `/testbed`. Five harnesses expose random identifiers and four expose dates, and six self-evolving systems expose such strings either always or only in some task setups (Appendix F). The audit shows exposure, not effect size. On their published benchmarks, GEPA, MIPROv2 and OPRO expose none of these strings (Agrawal et al., 2026; Opsahl-Ong et al., 2024; Yang et al., 2024), so our loop experiments add them (Section 5).

3 Comparisons Across Evaluator States

Section 2 shows that evaluator strings change responses. A keep/reject decision, however, compares two versions, which we call *candidates*, so what matters is how both respond to the same change.

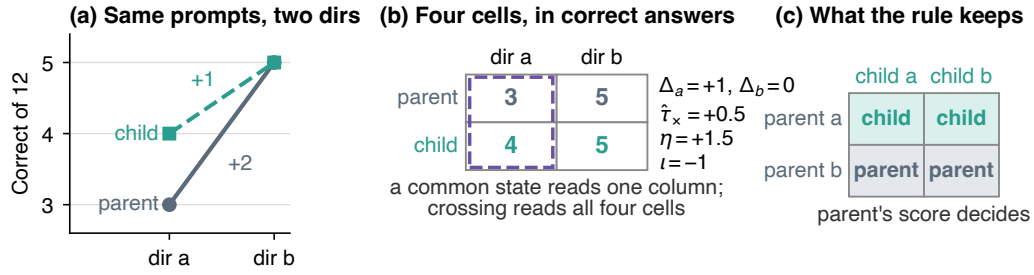


Figure 2: **A child can gain and still lose.** Its validation score rises, yet it loses the keep decision. (a) Correct answers of an archived parent–child pair, each held fixed across two directories, on the same 12 validation tasks. (b) The four cells and the decomposition of Equation 3. (c) The candidate that strict improvement keeps for each pairing of states: the parent's score, the loop's bar, decides.

Figure 2 shows an archived parent–child pair from the loops of Section 5 as a two-by-two table of two candidates by two states, a and b. A common-state comparison scores both candidates under one state. A sequential comparison may score the parent under a and the child under b. A crossed comparison uses all four cells.

3.1 Target and decomposition

Let $Y_t(c, s) \in \{0, 1\}$ be candidate c's grade on task t under evaluator state s, with $c \in \{\mathrm{P}, \mathrm{C}\}$ and state generator Q. Execution conditions such as tool clocks are held fixed; exact repeats check this, since greedy decoding alone does not guarantee it. When deployment draws fresh states, the target is

$$m_t(c) = \mathbb{E}_{s\sim Q} Y_t(c, s), \qquad \tau = |T|^{-1} \textstyle\sum_t \big[m_t(\mathrm{C}) - m_t(\mathrm{P})\big]. \tag{1}$$

Fixing one state measures the comparison at that state only, and repeating it exactly says nothing about other states drawn from Q.

A shared state cancels only shared responses. Write $Y_t(c, s) = m_t(c) + e_t(c, s)$, with variance $V_{t,c}$ and common-state covariance $\kappa_t = \mathrm{Cov}_Q(e_t(\mathrm{C}, s), e_t(\mathrm{P}, s))$. The contrast has state variance

$$\underbrace{V_{t,\mathrm{C}} + V_{t,\mathrm{P}}}_{\text{independent states}} \quad \text{or} \quad \underbrace{V_{t,\mathrm{C}} + V_{t,\mathrm{P}} - 2\kappa_t}_{\text{one state shared within the comparison}} . \tag{2}$$

Sharing a state helps only when the candidates respond together, which Section 4 measures. Fixing one state for every evaluation hides the rest rather than removing it: once the shared state changes, the candidate-specific part returns.

The crossed decomposition. With task means μ_{cs} under states a, b, let $\Delta_s = \mu_{\mathrm{C}s} - \mu_{\mathrm{P}s}$. The two sequential contrasts are $\hat{\Delta}_{ab} = \mu_{\mathrm{C}b} - \mu_{\mathrm{P}a}$ and $\hat{\Delta}_{ba} = \mu_{\mathrm{C}a} - \mu_{\mathrm{P}b}$. With $\hat{\tau}_\times = (\Delta_a + \Delta_b)/2$ and $\eta = \{(\mu_{\mathrm{P}b} - \mu_{\mathrm{P}a}) + (\mu_{\mathrm{C}b} - \mu_{\mathrm{C}a})\}/2$,

$$\hat{\Delta}_{ab} = \hat{\tau}_\times + \eta, \qquad \hat{\Delta}_{ba} = \hat{\tau}_\times - \eta, \qquad \iota = \Delta_b - \Delta_a. \tag{3}$$

This is the classical fold-over identity (Box & Hunter, 1961; Jones & Kenward, 2014). Here η is the state's main effect, the average of the two candidates' shifts, and ι the candidate-by-state interaction. Section 4 estimates both for every pair. In Figure 2, counting correct answers out of 12, $\hat{\tau}_\times = +0.5$, $\eta = +1.5$ and $\iota = -1$. When the state's average shift exceeds the true difference, $|\eta| > |\hat{\tau}_\times|$, the two sequential orders reach opposite decisions. Common-state decisions can disagree through ι even when $\eta = 0$.

Limits of single assignments. Without further assumptions, scoring the parent only under state a and the child only under b leaves the crossed effect in an interval 100 points wide, and rerunning those cells cannot narrow it (Proposition 1; Manski 1990). Search can also reward luck in the state: if K challengers are exact copies of the incumbent and scores are exchangeable without ties, keeping the best score picks a challenger with probability $K/(K+1)$ (Proposition 2; Smith & Winkler 2006). This limit illustrates the mechanism, not the rates we measure.

3.2 Study design

Four kinds of study follow: the channel audit of Section 2; crossed evaluations of archived candidates that no longer change; loops to which we add evaluator strings; and replays of recorded loops and budgets without new model calls. We use two *lineages*, each the recorded tree of candidates from one GEPA run on Qwen3-8B. Both use a fixed BIRD Mini-Dev split: 300 discovery tasks for training, 100 calibration tasks for validation and 100 test tasks. The lineages are *sealed*: their candidates were fixed before any state study. An *edge* is one parent–child pair, written $0{\to}2$ when candidate 2 descends from candidate 0, and an *edge run* is one run of an edge in one study. Lineage 1 has three edges and lineage 2 two. GEPA's final choice in each lineage is candidate 2, so we call $0{\to}2$ the *endorsed edge*. Each task is scored under two states of equal token length, and every cell runs twice so that the second run checks exact repetition. Appendix A maps the studies (Table 3), and the appendix guide defines the terms.

Before any outcome, we registered decision rules that asked whether the state changes lineage 1's keep/reject decisions on the full 100-task test set. No rule was triggered, so by these rules the state has a real effect that does not change decisions at this size. Each later study has its own registered plan, and lineage 2's registered runs also kept the same candidate. We call each registered plan a *freeze*: a self-timestamped file bound by hashes (Appendix A). The minibatch replays below are exploratory for lineage 1 and pre-specified for lineage 2. We report 95% intervals from a restricted wild cluster bootstrap that resamples the eleven databases (Cameron et al., 2008; MacKinnon & Webb, 2017; Webb, 2023). In a simulation with no true effect, its nominal 5% tests reject up to 8.4% of the time (Appendix M).

4 Fixed Candidates: Small Shifts, Unstable Decisions

4.1 State effects on grades

The average state shift is small. In the identifier study, η is 0, $+2$ and 0 accuracy points on lineage 1's three edges. With identically distributed identifiers, η is a mean-zero draw, not a bias. Across lineage 2's seven edge runs it ranges from -3.5 to $+4$ points. Some harnesses also number trial folders in the order candidates were found. In the frozen study that adds these discovery counters, on 400 tasks, the upper 95% limits of η are 1.0, 2.0 and 2.4 points, or up to 2.5 with the wild bootstrap. In that study, 42 of 4,800 cells failed the exact-repeat check (Appendix I).

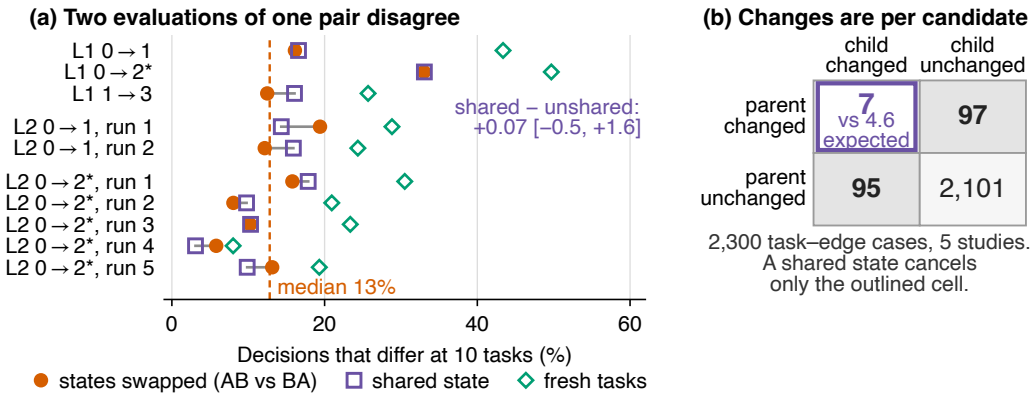


Figure 3: **Two evaluations of one pair often disagree, and sharing a state did not measurably help.** (a) Keep/reject decisions on the same 10-task minibatch that differ between two evaluations, for each of the ten edge runs (L1, L2: lineages; *: endorsed edge): with the two states swapped between evaluations, with one state shared within each evaluation, and with two fresh 10-task minibatches; exact reruns give 0%. The purple note gives the mean effect of sharing, in points, with a 95% interval. (b) Grade changes between states over 2,300 task–edge cases, with the count expected if the two candidates changed independently; a shared state cancels only the outlined cell.

Most changes belong to one candidate. Across lineage 1's studies, state changes alter about a quarter of each candidate's final SQL, ignoring whitespace, (Table 15), and on the 400 tasks they change 14–19 grades per candidate (Figure 8). Across the three lineage-1 studies, lineage 2's first four edge runs and the Qwen3-32B edge, 199 of 2,300 task–edge cases have a grade change. Only 7 of these involve both candidates, slightly more than the 4.6 expected if changes were independent (Figure 3b). The changes rarely coincide, so a shared state can cancel little of them (Equation 2).

4.2 Retention decisions

Decisions split, and sharing a state did not measurably help. Using the recorded grades, we apply the keep/reject rule twice to the same pair and minibatch, swapping the two states between the two evaluations (Appendix H). Across the ten identifier edge runs, three in lineage 1 and seven in lineage 2, two evaluations disagree on a median of 13% of decisions at 10 tasks (range 6–33%). Evaluations from two runs of lineage 2 reuse no state; post hoc, they disagree on 13% and 15% on its two edges, against 11% and 16% for the swap. Lineage 1's endorsed edge is the most sensitive, at 33% (database-cluster band 22–44%), and two fresh 10-task minibatches of it disagree on 50% (Figure 3a). Giving both candidates the same state changes the 10-task disagreement by -5 to $+4$ points across the ten edge runs, $+0.07$ on average. The ten edge runs come from five candidate pairs, and a post hoc bootstrap over these pairs gives a 95% interval of -0.5 to $+1.6$ points that describes only these pairs. The Qwen3-32B edge, graded on first queries, gives 23%, and its single-state verdict depends on the state (Appendix C).

Small changes flip boundary decisions. On ten tasks, one changed grade moves accuracy by ten points, so pairs with nearly equal scores often flip. Drawing different tasks moves decisions even more, and exact reruns capture neither source.

Runs cluster, but a run-level shift is not established. On lineage 2's endorsed edge, the two states of one run agree more often than states from different runs ($p < 0.0001$), yet the run's average contrast shows no reliable shift, and all five runs keep the parent. Between lineage 1's first two studies, whose states also differ in discovery counters, the kept candidate changes (Appendix I).

5 From Comparisons to an Optimization Loop

GEPA's own BIRD setup shows the model no paths. We add the audited Pi runner's two path lines, an instructions-file line and a working-directory line, to the system message, ahead of the prompt that GEPA evolves. We call these lines the *path block* and each run with them an *exposure arm*; directory names differ only in a suffix, v1, v2 and so on. In the small configuration, a child is admitted after beating its parent on one training task, the final choice uses 12 validation tasks, the budget is about 96 evaluation calls, and decoding is greedy. Each later stage was frozen before its outcomes were read; seven of the 16 grid runs had been read when the other nine were frozen (Appendix K). Three overlapping sets of runs recur below: a screen of the initial prompt in 123 directories, full loops in 18 of them at optimizer seed 0, and a grid over four of them at seeds 1–3. Table 2 summarizes eight settings.

5.1 Kept prompts across directories

The directory alone changes what the loop keeps. The path block does not edit the initial prompt, but it changes the prompt's scores: it moves its validation score from 5 of 12 to 2–5 across directories and raises its held-out accuracy from 44% to 48–54%. Comparing runs with and without the block therefore compares two harnesses, so we compare directories with one another. At first, SQL that reads the database clock broke exact repetition, so we reran these control runs with the clock fixed, under new freezes. A rerun in the same directory then repeats every evaluation call, with 12 or 100 validation tasks, while other directories diverge at the first or second call, in a model response, and keep different prompts (Table 2). By the registered rule, the directory name alone changes which candidate the loop keeps.

Low bars let children through. A child must strictly beat the initial prompt's validation score. Across 123 directories this bar ranges from 2 to 5 of 12, and 119 fall below the no-block value of 5

(Figure 10). A post hoc check in four other directories shows that the block lowers the initial prompt's score by 8–25 points on these 12 tasks but by 1–3 points on the other 88 tasks of the validation pool (Appendix K.3). The drop thus concentrates in this fixed 12-task set, which every screen uses. Where the bar is low, the loop keeps a child. Of the 18 directories run in full at optimizer seed 0, 13 of the 14 with a bar below 5 keep a child and 3 of the 4 with a bar of 5 keep the initial prompt; every kept child scores 3–16 points below it on held-out tasks (Figure 4a). These counts describe runs we chose, some by their bar, not deployed loops. A prediction registered for the second screen failed in one directory: at a bar of 5, v86 kept a child 7 points worse. A high bar lowers the risk but does not remove it. The losses thus come with the block, and the directory decides which worse child is kept. Without the block, GEPA keeps the same child at four optimizer seeds, 1 point better than the initial prompt.

5.2 Rescoring and larger settings

The kept children remain worse when rescored. Retested in two fresh directories, the children kept in v1–v3 at optimizer seed 0 remain 7–11 points below the initial prompt. Rescored without the path block, each of the nine distinct children scores below the initial prompt evaluated in the same job, by 1 to 9 points, and two of the nine 95% intervals exclude zero, one still after a Holm correction for nine tests (Figure 4b). These jobs ran on another GPU type, so we compare within jobs.

A larger validation set and other loops. With 100 validation tasks and a nominal 800 calls, all four directories lower the bar to 40–43 from 45 and keep children 3–11 points worse in their own directories. These children come from runs made before the clock fix. Rescored without the block, they score between 1 point below and 3 points above the initial prompt (Figure 4c). Of the two children from clock-fixed runs, 6 and 4 points worse under the block, one stays 7 points worse without it, with an interval excluding zero, and the other gains 2. At this scale the losses mostly depend on the block. An OPRO-style loop, which proposes new instructions from past scores, behaves alike. Its bar falls from 5 to 3–4, and every directory keeps an instruction 6–9 points worse, still 4–8 points worse in two fresh directories, while its run without the block gains 6. With Mistral-Small-24B the bar stays at 7–8 (7 without the block), and one of four directories keeps a child 2 points worse.

5.3 Selection rules and decision sizes

Fixed archives show the selector's sensitivity to the bar. Replaying the final choices of the 16 grid runs from their logs, raising only the initial prompt's final score to 5 of 12 undoes 9 of 12 harmful choices, and lowering it to 3 of 12 makes all 16 keep children (Figure 11c; Appendix L). This replay does not show how much of the search's loss the bar causes.

Small validation sets need large gaps. From the ten edge runs, we estimate how much a score varies with the state and with the task; this post hoc decision study sizes the gap a stable decision

Table 2: **Directories change what loops keep, and low bars go with worse children.** Adapted loops with the path block, on Qwen3-8B except the last row. Bar: the initial prompt's validation wins, with the value without the block in parentheses. Kept child: runs that keep a child (distinct prompts). Held-out change: kept child minus initial prompt on 100 test tasks in the run's own directory. No-block run: the loop without the block. Control: what a rerun in the same directory repeats (all records, that is, every evaluation call; only the kept prompt; or neither). Clock-fixed rows rerun directories of the row above them.

Loop, validation tasks	Runs	Bar: exposure (no-block run)	Kept child (distinct)	Held-out change own directory	No-block run kept − initial	Control: rerun repeats
GEPA, 12, grid	16	3–5 (5)	12 (7)	−15 to −6	+1	kept prompt
GEPA, 12, clock fixed	4	3–5 (5)	3 (3)	−14 to −11	+1	all records
GEPA, 12, new dirs	8	2–4 (5)	7 (4)	−16 to −3	–	–
GEPA, 12, bar screens	6	2–5 (5)	4 (3)	−13 to −7	–	–
GEPA, 100	4	40–43 (45)	4 (4)	−11 to −3	+3	neither
GEPA, 100, clock fixed	2	40–43 (45)	2 (2)	−6 to −4	–	all records
OPRO-style, 12	4	3–4 (5)	4 (2)	−9 to −6	+6	all records
GEPA, 12, Mistral-Small	4	7–8 (7)	1 (1)	−2	0	kept prompt

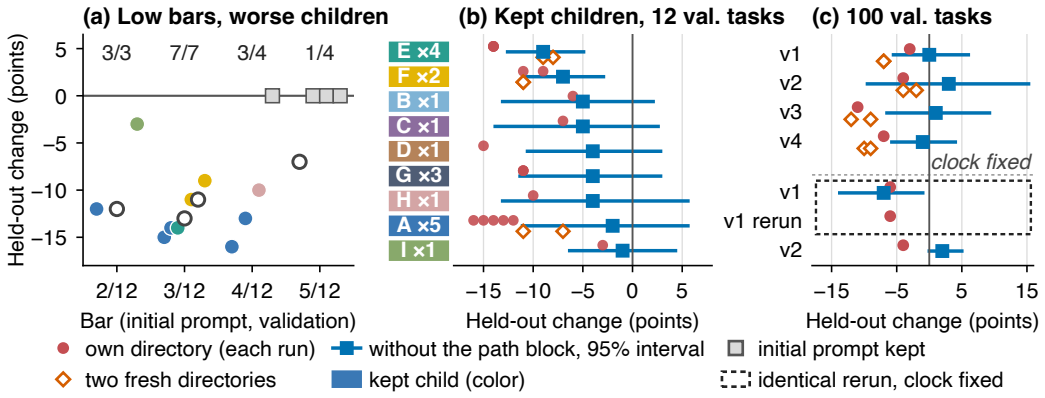


Figure 4: **Low bars go with worse children, and the directory decides which.** (a) One point per directory at optimizer seed 0 (v1–v12 and six screened ones), 12 validation tasks: held-out change of the kept prompt by the bar; colors mark the children of panel (b), open circles children not rescored, squares the initial prompt; numbers count directories that keep a child. (b) The nine children kept in v1–v12, labeled A–I as in Figure 9, in their own directories and without the path block, and the three kept at seed 0 in two fresh directories; $\times n$ counts runs. (c) The same at 100 validation tasks, before and after the clock was fixed; clock-fixed children were rescored on the H200.

needs. Under a normal approximation, two evaluations on 12 tasks agree on 90% of decisions only when the candidates differ by about 11 points, or 19 points when tasks are redrawn too. At 100 tasks the gaps are 4 and 7 points (Figure 5a). With independently sampled tasks and states, n tasks and m states per task give $\mathrm{Var}(\hat{\tau}) = V_{\mathrm{task}}/n + V_{\mathrm{state}}/(nm)$. At a fixed budget $B = nm$, drawing m states per task instead of one adds $(m-1)V_{\mathrm{task}}/B$, so a fixed budget is best spent on more tasks with one fresh state each; a finite replay of the ten edge runs agrees (Appendix M). These rough guides ignore ties and treat states as drawn within a run; run clustering would widen them.

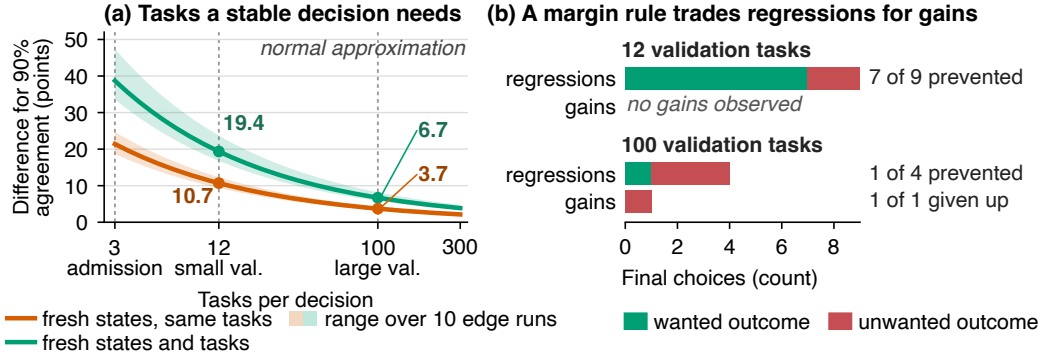


Figure 5: **Small decisions need large gaps, and a margin rule is not free.** (a) Accuracy difference at which two independent evaluations agree on 90% of strict-improvement decisions (normal approximation, post hoc; median and range over the ten edge runs), with fresh states on the same tasks or with fresh tasks too. (b) Final choices under the registered margin rule, two tasks at 12 validation tasks and four at 100: green marks regressions prevented, red marks regressions kept or gains lost.

Margin rules, untested in live search. A stable selector need not be useful: always keeping the parent agrees perfectly and misses every improvement. A margin rule keeps a child at the final choice only if it wins by a set number of tasks. We chose a two-task margin after reading all but one of the 16 grid and 4 OPRO runs; in them it undoes 7 of 16 harmful choices. Registered before the remaining runs, it blocks 7 of 9 regressions in all 15 loops with 12 validation tasks and meets its registered

criterion (Figure 5b), but these runs contain no gain to lose. At 100 validation tasks, a four-task margin blocks one of four regressions and gives up the only gain, so the criterion fails.

6 Related Work

Prompt sensitivity and rerun noise. Prompt formats and small perturbations change model outputs, which motivates multi-prompt evaluation and error bars (Salinas & Morstatter, 2024; Sclar et al., 2024; Polo et al., 2024; Mizrahi et al., 2024; Miller, 2024; Chen et al., 2026), and irrelevant context distracts models (Shi et al., 2023). In these studies the experimenter chooses the perturbation. A practitioner report warns that harness variables such as paths can sway agent evaluations (Mastykarz, 2026); we measure such strings and show why exact reruns cannot reveal them. In systems research, the size of the UNIX environment, which changes with the working directory, biases measured speedups (Mytkowicz et al., 2009), and randomizing memory layout restores sound comparisons (Curtsinger & Berger, 2013). Reruns can detect nondeterministic serving (Atil et al., 2024; Yuan et al., 2025; He & Thinking Machines Lab, 2025; Song et al., 2025; Zhang et al., 2025) and other reproducibility problems of agent evaluations (Kapoor et al., 2025; 2026; Zhu et al., 2025), but an exact rerun by design reproduces the evaluator state.

Noisy adaptive selection. Prompt and agent optimizers keep candidates based on evaluation scores (Yang et al., 2024; Hu et al., 2025; Zhang et al., 2026; Agrawal et al., 2026). Budget matching, retention and credit assignment affect their reported gains (Wang et al., 2026; Luo et al., 2026; Liu et al., 2026), and racing, uncertainty handling, anytime-valid tests for self-evolving agents such as PACE, and reusable holdouts address noisy selection (Hutter et al., 2009; López-Ibáñez et al., 2016; Hansen et al., 2009; Shawn, 2026; Dwork et al., 2015). These guarantees cover only the variation that evaluations resample, not evaluator-written context that an exact rerun holds fixed. Our margin rule resembles the Ladder leaderboard's threshold (Blum & Hardt, 2015).

Measurement design. Seeds, task splits and harness choices contribute to benchmark variation (Henderson et al., 2018; Bouthillier et al., 2019; Dodge et al., 2020; Bouthillier et al., 2021; Madaan et al., 2024; Heineman et al., 2025; Yi & Song, 2026). Our advice to redraw conditions follows Bouthillier et al. (2021) and Sclar et al. (2024); pass^k (Yao et al., 2025) and power analyses (Card et al., 2020) address repeated trials and sample size. Generalizability designs appear in information retrieval (Voorhees & Buckley, 2002; Bodoff & Li, 2007; Urbano et al., 2013) and LLM evaluation (Messing, 2026; Wang, 2025; Keller et al., 2026). Common random numbers motivate pairing, that is, giving both candidates the same state (Glasserman & Yao, 1992); we show where pairing stops helping. We use repeatability and replicability operationally (National Academies of Sciences, Engineering, and Medicine, 2019; Joint Committee for Guides in Metrology, 2012).

7 Discussion and Conclusion

Reporting. State whether the evaluator state is redrawn per task or per run, and replicate comparisons under fresh states rather than exact reruns. With per-task states, spend a fixed budget on more tasks, one fresh state each. When a whole run shares one state, replicate whole runs, because more tasks cannot estimate run-to-run variation. Keep telemetry such as `elapsed_s` out of model inputs.

Limitations. The executable evidence concerns one benchmark, BIRD. The archived GEPA runs use Qwen3-8B, and the 32B and hosted studies stop at the first query (Appendices C and D). Our loops place the path block before the prompt, whereas Pi appends it, and decode greedily. Small loops admit a child after one training task and select on a fixed set of 12 validation tasks; published GEPA uses three tasks and samples at temperature 0.6. The loops adapt GEPA and OPRO rather than reproduce their published setups. Harness choice matters too: lineage 2's chosen child gains 1 point in GEPA's own setup but loses 7.5 to 15.5 points in the Pi harness. Our 95% intervals may cover less than 95% (Appendix M) and describe only the lineages studied. Disagreement counts changed decisions, not their cost. With the block, our loops never keep a better prompt.

An exact rerun asks whether a measurement repeats at the state it preserves. A self-evolving loop must also know whether its comparison survives redrawn states, which replication can show, and whether the kept candidate helps, which our loops leave open.

Reproducibility Statement

Sections 3.2 and 5 and the appendices specify tasks, state assignments, budgets, recoveries and endpoints. Upon publication, we will release the frozen plans, hashes, amendments, aggregate audits and analysis code. One script per figure and table reads only these summaries, and post hoc analyses are stored under their own identifiers. Freezes are self-timestamped; Appendix A distinguishes original predictions from later analyses. Public weights and benchmark data are not redistributed. Model responses and trajectories are not redistributed either; this code regenerates them from the public weights and benchmark, except in the hosted-model study, whose endpoints may change. Exact repeats need not transfer across kernels or providers.

AI Use Statement

We used LLMs in several parts of this work. For the manuscript, the authors supplied the draft, and LLMs polished the text and suggested its structure. For the experiments, the authors designed the experiments and LLMs wrote the code; the authors then ran the experiments and analyzed the results. LLMs also helped search and verify the literature. The authors take full responsibility for the paper.

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, Alexandros G. Dimakis, Ion Stoica, Dan Klein, Matei Zaharia, and Omar Khattab. GEPA: Reflective prompt evolution can outperform reinforcement learning. In *International Conference on Learning Representations*, pp. 8479–8565, 2026. URL https://openreview.net/forum?id=RQm2KQTM5r.

Aider-AI. Aider. https://github.com/Aider-AI/aider, 2026. Source code at commit 5dc9490bb35f, accessed 2026-09-25.

Algorithmic Superintelligence. OpenEvolve. https://github.com/algorithmicsuperintelligence/openevolve, 2026. Source code at commit 411fb59c886c, accessed 2026-09-25.

Berk Atil, Sarp Aykent, Alexa Chittams, Lisheng Fu, Rebecca J. Passonneau, Evan Radcliffe, Guru Rajan Rajagopal, Adam Sloan, Tomasz Tudrej, Ferhan Ture, Zhe Wu, Lixinyu Xu, and Breck Baldwin. Non-determinism of "deterministic" LLM settings. arXiv:2408.04667, 2024. URL https://arxiv.org/abs/2408.04667.

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2-Bench: Evaluating conversational agents in a dual-control environment. In *International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=OC2z7iSQKa.

Avrim Blum and Moritz Hardt. The ladder: A reliable leaderboard for machine learning competitions. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 1006–1014, 2015.

David Bodoff and Pu Li. Test theory for assessing IR test collections. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 367–374, 2007. doi: 10.1145/1277741.1277805.

Xavier Bouthillier, César Laurent, and Pascal Vincent. Unreproducible research is reproducible. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 725–734, 2019.

Xavier Bouthillier, Pierre Delaunay, Mirko Bronzi, Assya Trofimov, Brennan Nichyporuk, Justin Szeto, Naz Sepah, Edward Raff, Kanika Madan, Vikram Voleti, Samira Ebrahimi Kahou, Vincent Michalski, Dmitriy Serdyuk, Tal Arbel, Chris Pal, Gaël Varoquaux, and Pascal Vincent. Accounting for variance in machine learning benchmarks. In *Proceedings of Machine Learning and Systems*, volume 3, pp. 747–769, 2021. URL https://proceedings.mlsys.org/paper_files/paper/2021/hash/0184b0cd3cfb185989f858a1d9f5c1eb-Abstract.html.

G. E. P. Box and J. S. Hunter. The 2^{k-p} fractional factorial designs, Part I. *Technometrics*, 3(3): 311–351, 1961. doi: 10.1080/00401706.1961.10489951.

Robert L. Brennan. *Generalizability theory*. Statistics for Social Science and Public Policy. Springer, New York, 2001. doi: 10.1007/978-1-4757-3456-0.

A. Colin Cameron, Jonah B. Gelbach, and Douglas L. Miller. Bootstrap-based improvements for inference with clustered errors. *Review of Economics and Statistics*, 90(3):414–427, 2008. doi: 10.1162/rest.90.3.414.

Dallas Card, Peter Henderson, Urvashi Khandelwal, Robin Jia, Kyle Mahowald, and Dan Jurafsky. With little power comes great responsibility. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9263–9274, 2020. doi: 10.18653/v1/2020.emnlp-main.745.

Ke Chen, Xucheng Yu, Yufei Zhou, and Haohan Wang. Prompt stability matters: Evaluating and optimizing auto-generated prompt in general-purpose systems. In *Proceedings of the Conference on Parsimony and Learning*, volume 328 of *Proceedings of Machine Learning Research*, pp. 360–374, 2026. URL https://proceedings.mlr.press/v328/chen26b.html.

Lee J. Cronbach, Goldine C. Gleser, Harinder Nanda, and Nageswari Rajaratnam. *The dependability of behavioral measurements: Theory of generalizability for scores and profiles*. John Wiley & Sons, New York, 1972.

Charlie Curtsinger and Emery D. Berger. STABILIZER: Statistically sound performance evaluation. In *Proceedings of the 18th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, pp. 219–228, 2013. doi: 10.1145/2451116.2451141.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*, 2020.

DSPy. DSPy documentation: Cache. https://dspy.ai/tutorials/cache/, 2026. Accessed 2026-09-25.

Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. The reusable holdout: Preserving validity in adaptive data analysis. *Science*, 349(6248):636–638, 2015.

Esakkivel Esakkiraja, Denis Akhiyarov, Vikas Yadav, Sai Rajeswar, Patrice Bechard, Sridhar Nemala, and Sagar Davasam. StarHarness: Evolving Harnesses with Stratified Search for Enterprise Environments. arXiv:2608.24804, 2026. URL https://arxiv.org/abs/2608.24804.

Paul Glasserman and David D. Yao. Some guidelines and guarantees for common random numbers. *Management Science*, 38(6):884–908, 1992. doi: 10.1287/mnsc.38.6.884.

Google. Gemini CLI. https://github.com/google-gemini/gemini-cli, 2026. Source code at commit bedef96ef429, accessed 2026-09-25.

Nikolaus Hansen, André S. P. Niederberger, Lino Guzzella, and Petros Koumoutsakos. A method for handling uncertainty in evolutionary optimization with an application to feedback control of combustion. *IEEE Transactions on Evolutionary Computation*, 13(1):180–197, 2009. doi: 10.1109/TEVC.2008.924423.

Harbor Framework. Terminal-Bench and the Terminus agent. https://github.com/harbor-framework/terminal-bench-1, 2026. Source code at commit d28711d0da26, accessed 2026-09-25.

Horace He and Thinking Machines Lab. Defeating nondeterminism in LLM inference. Thinking Machines Lab: Connectionism, 2025.

David Heineman, Valentin Hofmann, Ian Magnusson, Yuling Gu, Noah A. Smith, Hannaneh Hajishirzi, Kyle Lo, and Jesse Dodge. Signal and noise: A framework for reducing uncertainty in language model evaluation. arXiv:2508.13144, 2025.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pp. 3207–3214, 2018. doi: 10.1609/aaai.v32i1.11694.

Shengran Hu, Cong Lu, and Jeff Clune. Automated design of agentic systems. In *International Conference on Learning Representations*, pp. 21344–21377, 2025. URL https://openreview.net/forum?id=t9U3LW7JVX.

Frank Hutter, Holger H. Hoos, Kevin Leyton-Brown, and Thomas Stützle. ParamILS: An automatic algorithm configuration framework. *Journal of Artificial Intelligence Research*, 36:267–306, 2009. doi: 10.1613/jair.2861.

Joint Committee for Guides in Metrology. International vocabulary of metrology – basic and general concepts and associated terms (VIM), 3rd edition. JCGM 200:2012, 2012.

Byron Jones and Michael G. Kenward. *Design and analysis of cross-over trials*. Chapman and Hall/CRC, third edition, 2014. doi: 10.1201/b17537.

Sayash Kapoor, Benedikt Stroebl, Zachary S. Siegel, Nitya Nadgir, and Arvind Narayanan. AI agents that matter. *Transactions on Machine Learning Research*, 2025. URL https://openreview.net/forum?id=Zy4uFzMviZ.

Sayash Kapoor, Benedikt Stroebl, Peter Kirgis, Nitya Nadgir, Zachary S. Siegel, Boyi Wei, Tianci Xue, Ziru Chen, Felix Chen, Saiteja Utpala, Franck Ndzomga, Dheeraj Oruganty, Sophie Luskin, Kangheng Liu, Botao Yu, Amit Arora, Dongyoon Hahm, Harsh Trivedi, Huan Sun, Juyong Lee, Tengjun Jin, Yifan Mai, Yifei Zhou, Yuxuan Zhu, Rishi Bommasani, Daniel Kang, Dawn Song, Peter Henderson, Yu Su, Percy Liang, and Arvind Narayanan. Holistic Agent Leaderboard: The missing infrastructure for AI agent evaluation. In *International Conference on Learning Representations*, pp. 98778–98849, 2026. URL https://openreview.net/forum?id=vUaY1t64ZZ.

Andrew Keller, Kweku Kwegyir-Aggrey, Ryan Steed, Anita Rao, Julia Sharp, and Amanda Bergman. Expanding the AI evaluation toolbox with statistical models. Technical Report NIST AI 800-3, National Institute of Standards and Technology, 2026.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. DSPy: Compiling declarative language model calls into state-of-the-art pipelines. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=sY5N0zY5Od.

Jinyang Li, Binyuan Hui, Ge Qu, Jiaxi Yang, Binhua Li, Bowen Li, Bailin Wang, Bowen Qin, Rongyu Cao, Ruiying Geng, Nan Huo, Xuanhe Zhou, Chenhao Ma, Guoliang Li, Kevin C. C. Chang, Fei Huang, Reynold Cheng, and Yongbin Li. Can LLM already serve as a database interface? A BIg bench for large-scale database grounded text-to-SQLs. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2305.03111.

Yuxuan Liu, Zhaochen Su, Yuhao Zhang, Jiahe Guo, Zhongwei Xie, Huihao Jing, Lingyun Xie, Qing Zong, Yauwai Yim, Zhixiong Zhang, Haoran Li, and Yangqiu Song. Rethinking self-evolving agent skills: Feedback dynamics over multiple rounds. arXiv:2608.02636, 2026. URL https://arxiv.org/abs/2608.02636.

Manuel López-Ibáñez, Jérémie Dubois-Lacoste, Leslie Pérez Cáceres, Mauro Birattari, and Thomas Stützle. The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, 3:43–58, 2016. doi: 10.1016/j.orp.2016.09.002.

Xiaotian Luo, Dizhan Xue, Fengxingyu Wang, Chuanrui Hu, and Yafeng Deng. HarnessBank: Semantic Gene-Bank Search with Gated Verification for Agent-Harness Self-Evolution. arXiv:2607.13683, 2026. URL https://arxiv.org/abs/2607.13683.

James G. MacKinnon and Matthew D. Webb. Wild bootstrap inference for wildly different cluster sizes. *Journal of Applied Econometrics*, 32(2):233–254, 2017. doi: 10.1002/jae.2508.

Lovish Madaan, Aaditya K. Singh, Rylan Schaeffer, Andrew Poulton, Sanmi Koyejo, Pontus Stenetorp, Sharan Narang, and Dieuwke Hupkes. Quantifying variance in evaluation benchmarks. arXiv:2406.10229, 2024.

Charles F. Manski. Nonparametric bounds on treatment effects. *American Economic Review*, 80(2): 319–323, 1990. Papers and Proceedings.

Waldek Mastykarz. The hidden variables in your agent eval. Microsoft for Developers, July 2026. URL https://developer.microsoft.com/blog/the-hidden-variables-in-your-agent-eval/. accessed 2026-09-18.

Solomon Messing. Hidden measurement error in LLM pipelines distorts annotation, evaluation, and benchmarking. arXiv:2604.11581, 2026. URL https://arxiv.org/abs/2604.11581.

Evan Miller. Adding error bars to evals: A statistical approach to language model evaluations. arXiv:2411.00640, 2024. URL https://arxiv.org/abs/2411.00640.

Moran Mizrahi, Guy Kaplan, Dan Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. State of what art? A call for multi-prompt LLM evaluation. *Transactions of the Association for Computational Linguistics*, 12:933–949, 2024. doi: 10.1162/tacl_a_00681.

Todd Mytkowicz, Amer Diwan, Matthias Hauswirth, and Peter F. Sweeney. Producing wrong data without doing anything obviously wrong! In *Proceedings of the 14th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, pp. 265–276, 2009. doi: 10.1145/1508244.1508275.

National Academies of Sciences, Engineering, and Medicine. *Reproducibility and replicability in science*. The National Academies Press, Washington, DC, 2019. doi: 10.17226/25303.

OpenAI. Codex CLI. https://github.com/openai/codex, 2026. Source code at commit 8ae55c863db2, accessed 2026-09-25.

OpenHands. OpenHands software agent SDK and benchmarks. https://github.com/OpenHands/software-agent-sdk, 2026. Source code at commit 0620c8b1cd6b, accessed 2026-09-25.

Krista Opsahl-Ong, Michael J. Ryan, Josh Purtell, David Broman, Christopher Potts, Matei Zaharia, and Omar Khattab. Optimizing instructions and demonstrations for multi-stage language model programs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 9340–9366, Miami, Florida, USA, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.525.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. YaRN: Efficient context window extension of large language models. In *International Conference on Learning Representations*, pp. 31932–31951, 2024. URL https://openreview.net/forum?id=wHBfxhZu1u.

Felipe Maia Polo, Ronald Xu, Lucas Weber, Mírian Silva, Onkar Bhardwaj, Leshem Choshen, Allysson Flavio Melo de Oliveira, Yuekai Sun, and Mikhail Yurochkin. Efficient multi-prompt evaluation of LLMs. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2405.17202.

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

Sakana AI. ShinkaEvolve. https://github.com/SakanaAI/ShinkaEvolve, 2026. Source code at commit 9912af12d423, accessed 2026-09-25.

Abel Salinas and Fred Morstatter. The butterfly effect of altering prompts: How small changes and jailbreaks affect large language model performance. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 4629–4651, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.275.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. Quantifying language models' sensitivity to spurious features in prompt design or: How I learned to start worrying about prompt formatting. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.11324.

Zayx Shawn. PACE: Anytime-Valid Acceptance Tests for Self-Evolving Agents. arXiv:2606.08106, 2026. URL https://arxiv.org/abs/2606.08106.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H. Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 31210–31227, 2023.

James E. Smith and Robert L. Winkler. The optimizer's curse: Skepticism and postdecision surprise in decision analysis. *Management Science*, 52(3):311–322, 2006. doi: 10.1287/mnsc.1050.0451.

Yifan Song, Guoyin Wang, Sujian Li, and Bill Yuchen Lin. The good, the bad, and the greedy: Evaluation of LLMs should not ignore non-determinism. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4195–4206, Albuquerque, New Mexico, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.naacl-long.211.

SWE-agent. mini-SWE-agent. https://github.com/SWE-agent/mini-swe-agent, 2026a. Source code at commit 04d809ceab9d, accessed 2026-09-25.

SWE-agent. SWE-agent. https://github.com/SWE-agent/SWE-agent, 2026b. Source code at commit 3ea751c087f3, accessed 2026-09-25.

Julián Urbano, Mónica Marrero, and Diego Martín. On the measurement of test collection reliability. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 393–402, 2013. doi: 10.1145/2484028.2484038.

Ellen M. Voorhees and Chris Buckley. The effect of topic set size on retrieval experiment error. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 316–323, 2002. doi: 10.1145/564376.564432.

Sida Wang. Measuring all the noises of LLM evals. arXiv:2512.21326, 2025. URL https://arxiv.org/abs/2512.21326.

Yike Wang, Huaisheng Zhu, Zhengyu Hu, Yige Yuan, Zhengyu Chen, Shakti Senthil, Hannaneh Hajishirzi, Yulia Tsvetkov, Pradeep Dasigi, and Teng Xiao. Rethinking the evaluation of harness evolution for agents. arXiv:2607.12227, 2026. URL https://arxiv.org/abs/2607.12227.

Matthew D. Webb. Reworking wild bootstrap-based inference for clustered errors. *Canadian Journal of Economics/Revue canadienne d'économique*, 56(3):839–858, 2023. doi: 10.1111/caje.12661.

Jinghan Xu, Yikai Zhang, Aili Chen, Weiyuan Li, Jiaqing Liang, and Deqing Yang. Verify smarter, evolve further: Efficient harness evolution through behavior-aware verification. arXiv:2608.27311, 2026. URL https://arxiv.org/abs/2608.27311.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. arXiv:2505.09388, 2025. URL https://arxiv.org/abs/2505.09388.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. In *International Conference on Learning Representations*, pp. 12028–12068, 2024. URL https://openreview.net/forum?id=Bb4VGOWELI.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. In *International Conference on Learning Representations*, pp. 9965–10017, 2025. URL https://openreview.net/forum?id=roNSXZpUDN.

Haiwen Yi and Xinyuan Song. Measuring harness-induced belief divergence in multi-step LLM agents. arXiv:2607.04528, 2026.

Jiayi Yuan, Hao Li, Xinheng Ding, Wenya Xie, Yu-Jhe Li, Wentian Zhao, Kun Wan, Jing Shi, Xia Hu, and Zirui Liu. Give me FP32 or give me death? challenges and solutions for reproducible reasoning. In *Advances in Neural Information Processing Systems*, 2025. arXiv:2506.09501.

Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Pan Lu, Zhi Huang, Carlos Guestrin, and James Zou. Optimizing generative AI by backpropagating language model feedback. *Nature*, 639(8055):609–616, 2025. doi: 10.1038/s41586-025-08661-4.

Mario Zechner. Pi: A coding agent CLI. https://github.com/earendil-works/pi, 2026. Package @earendil-works/pi-coding-agent, version 0.80.10, accessed 2026-09-25.

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*, pp. 104223–104294, 2026. URL https://openreview.net/forum?id=pUpzQZTvGY.

Tuo Zhang, Jinyue Yuan, and Salman Avestimehr. Revisiting OPRO: The limitations of small-scale LLMs as optimizers. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 1727–1735, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.100.

Ziyang Zhang, Xinheng Ding, Jiayi Yuan, Rixin Liu, Huizi Mao, Jiarong Xing, and Zirui Liu. Deterministic inference across tensor parallel sizes that eliminates training-inference mismatch. arXiv:2511.17826, 2025. URL https://arxiv.org/abs/2511.17826.

Tailin Zhou. Hierarchical self-improvement: A framework for task-specific evolvable agent harnesses. arXiv:2608.08466, 2026. URL https://arxiv.org/abs/2608.08466.

Yuxuan Zhu, Tengjun Jin, Yada Pruksachatkun, Andy Zhang, Shu Liu, Sasha Cui, Sayash Kapoor, Shayne Longpre, Kevin Meng, Rebecca Weiss, Fazl Barez, Rahul Gupta, Jwala Dhamala, Jacob Merizian, Mario Giulianelli, Harry Coppock, Cozmin Ududec, Antony Kellermann, Jasjeet Sekhon, Jacob Steinhardt, Sarah Schwettmann, Arvind Narayanan, Matei A. Zaharia, Ion Stoica, Percy Liang, and Daniel Kang. Establishing best practices in building rigorous agentic benchmarks. In *Advances in Neural Information Processing Systems*, volume 38, pp. 184435–184475, 2025. doi: 10.52202/085713-5547.

APPENDIX GUIDE

The appendices keep failed predictions, recoveries and complete study scopes, and they label analyses introduced after outcomes. Appendix A documents every study and frozen prediction; Appendices B–F support Section 2, Appendix G Section 3, Appendices H–J Section 4, and Appendices K–M Section 5. The table below defines terms.

Term	Meaning in this paper
Evaluator state	Strings an evaluator writes into model inputs: a run directory, an identifier, a latency field
Facet	A measurement condition a score may need to generalize over
Repeatability; replicability	Agreement at one fixed state; agreement under fresh states
Lineage; sealed	One GEPA run's recorded candidate tree; frozen before any state study
Edge; endorsed edge	A parent–child pair; the pair GEPA selects in a lineage, 0→2
Edge run	One run of one edge in one study; Section 4 uses ten
Crossed comparison	Both candidates under both states; η is the state's main effect and ι the candidate-by-state interaction (Equation 3)
Bar	The initial prompt's validation score, which a child must strictly beat
Sequential (ordinary) assignment	Parent and child under different states
Record	One evaluation call (GEPA's metric call) and its episode in a loop's training log
Path block	The Pi runner's two path lines, added to a loop's system message
Exposure arm; no-block run; A/A arm	A loop run with the block; the same loop without it; a rerun showing the same directory string, used as a control
Initial prompt; optimizer seed	The loop's starting candidate; the random seed of GEPA or OPRO
Frozen	Fixed before outcomes in a self-timestamped file bound by hashes
S1–S3; Stage N	Lineage 1's three studies; later frozen stages (Tables 5 and 6): 2a–2b lineage-2 runs; 4–6 loop settings (4c Mistral, 4d seed grid, 4e more directories, 5 OPRO, 6 100 validation tasks); 7 hosted models; 8, 11 rescoring without the block; 9a, 10a clock-fixed controls; 9b no-block seeds; 9c, 10b high-bar screens

A STUDY PROTOCOLS

Table 3: **Studies and what each can establish.** The kinds of evidence answer different questions and do not replicate one another.

Study	Varied condition	Units	Endpoint; status
Native channel (§2)	Path identifier; latency field	60 tasks × 4 Qwen3 settings; 100 Mistral-7B tasks; 60 episodes	Response, grade, trajectory; frozen
Lineage 1, S1–S3 (§4)	Identifiers; counters in S2, S3	3 edges; 100 test or 400 training tasks	Contrast, decision; exploratory
Lineage 2, Stages 2a–2b (§4)	Run string; identifiers	2 edges; 100 test tasks; 7 edge runs	As lineage 1, plus run clustering; frozen
Adapted loops, Stages 4–11 (§5)	Directory in an added path block; optimizer seed	GEPA (12 or 100 validation tasks), OPRO, Mistral-Small-24B; 123 screened directories	Kept prompt, held-out accuracy; frozen per stage
Replays (§5)	Final parent score; evaluation allocation	16 archives; 10 edge runs	Final choice, regret; post hoc, no model calls

Tasks. BIRD Mini-Dev has 500 questions over 11 databases, split into 300 discovery questions used for training, 100 calibration questions used for validation and 100 test questions. The 60-task mechanism studies use two cohorts of 30 discovery tasks; the Mistral confirmation uses 100 further discovery tasks. None of the seven candidate instructions contains a 20-hex string, path, `AGENTS.md`, run directory, timing field or counter.

Table 4: Crossed studies of real GEPA edges. Each run crosses parent and child over two fresh equal-token states per task, with an exact repeat. Exploratory: specified after the study was read. S3's freeze barred candidate contrasts on its in-sample tasks; reporting their state component only is a deviation.

Study	Lineage, edges	Tasks	Episodes	States a, b differ in	Analysis status
S1	1: $0\to1$, $0\to2$, $1\to3$	100 test	2,400	identifier	frozen; facet exploratory
S2	1: same	100 test	2,400	identifier; counter	frozen; facet exploratory
S3	1: same	400 train	9,600	identifier; counter	frozen bias bound; contrasts a deviation
Stage 2a	2: $0\to1$, $0\to2$	100 test	3,200	identifier	frozen before outcomes
Stage 2b	2: $0\to2$	100 test	2,400	identifier	frozen before outcomes
32B edge	32B: one edge	100 test	800 calls	identifier	frozen; first turn only

Table 5: Frozen predictions and outcomes: crossed studies and hosted models. Freezes cover their new arms only. Prior exposure and amendments are disclosed below; lineage-1 facet analyses and loop intervals are exploratory or post hoc.

Study	Frozen prediction or rule	Outcome
S1	a trigger on some edge: selectors disagree, $\lvert\eta\rvert \geq \lvert\tau\rvert$, or the worst state rejects an ordinary winner; if none, the effect is read as real but not consequential for retention	none fired; later studies are new freezes
S2	the same triggers; if none, a stronger negative for the shared pathway	none fired
S2, bias	$\lvert\eta\rvert$ materially above S1's 0.0–2.0 points	η $+1.0$ to $+2.5$ on all edges; not material; no interval excludes zero
S3	some edge's η interval excludes zero, with η in 0.5–3.5 points; else a bound	none excludes zero; bounds 1.0, 2.0, 2.4 points
Stage 2a	the kept candidate can change between runs	both runs keep the parent on both edges; lineage 1's change not reproduced
Stage 2b	$\lvert r\rvert < 0.3$ in every run; one-sided tests of H1 run clustering, H2 mean shift, H3 whole-run effect	r from -0.25 to $+0.19$ in seven edge runs; H1 rejected on $0\to2$ ($p < 0.0001$), not on $0\to1$; H2, H3 not rejected
Stage 7	at temperature 0 the state adds to repeat noise for both models	held for DeepSeek, failed for GLM

First turn and complete episodes. The first-turn study calls each of 60 tasks under identifier A, an equal-token identifier B and an exact repeat of A, for two Qwen sizes and two context configurations, and grades the first parseable SQL; unparsable output scores zero (cohorts in Table 8). Two further path variants ran but are not analyzed here. The complete-episode studies run three episodes per task (A, B, repeat of A) with at most 30 tool steps. After the raw study's repeat failure, a new prospective run removed only the top-level `elapsed_s` from model-facing tool messages; timing stays in the native log. The clock task stays in the run but, as prespecified, not in the denominator of 59.

Identical copies. Before opening the latency-removed outcomes, we fixed five blocks of 11 or 12 of the 59 non-clock tasks and treated the runs under identifiers A and B as two byte-identical copies.

Repeatability of the crossed runs. S3's frozen exact-repeat gate failed: 42 of 4,800 cells did not repeat their complete trajectory. As frozen, such cells are recorded, not refused, and all statistics use repeat 1. Replayed S3 shards reproduced 153 of 153 and 139 of 139 responses byte for byte on other cards and days. The 18-turn limit ended 13 to 25 percent of episodes. All analyses use the auditor's independent regrade. It differs from the in-episode reward, which GEPA optimized, on 16, 14 and 10 of 3,200 S3 episodes per edge, all floating-point ties.

Crossed runs of real edges (S1). Table 4 lists the crossed studies. Each edge's plan binds tasks, candidate bytes, two unseen 20-hex states per task, two exact repeats, a cyclic Latin order, an 18-turn

Table 6: Frozen predictions and outcomes: adapted loops, in stage order (conventions as in Table 5). E1–E5: loop endpoints defined in Appendix K.1.

Study	Frozen prediction or rule	Outcome
Stage 4	E1, E2 repeat every record; two or more kept prompts	choices repeat, records do not (clock); four prompts
Stage 4c	as Stage 4, Mistral-Small-24B	E1 holds; E2 repeats choices only; two prompts
Stage 4d	the bar is invariant across optimizer seeds	held
Stage 4e	every new kept child is worse; low bars keep more children	7/7 worse; no new high bar, second prediction unresolved
Stage 5	E2 repeats every record; two or more kept prompts	E1 canceled; E2 held; two prompts, both below the initial prompt in E5
Stage 6	the A/A control repeats decisions; directories diverge early	divergence held; the control changed its choice at record 137, so attribution failed
Stage 8	without the block, every kept child scores below the initial prompt	held for all 9 children at 12 tasks; failed for 3 of 5 at 100
Stage 9a	clock fixed: both controls repeat every record and directories diverge early	held; divergence at record 1–2; attribution met
Stage 9b	count no-block optimizer seeds that keep a worse child	0 of 3 (each keeps the seed-0 child, $+1$)
Stage 9c	bar-5 directories keep the initial prompt, bar ≤ 3 a worse child; needs two at bar 5	1 of 36 at bar 5: untestable; it keeps the initial prompt, its partner a worse child
Stage 10a	clock fixed, 100 tasks: the A/A rerun repeats every record; v1 and v2 diverge within 10 records and keep different prompts	all held (813 records repeat); children 6 and 4 points worse
Stage 10b	the Stage 9c prediction in 75 new directories	testable (2 at bar 5); failed: v64 keeps the initial prompt, v86 a child 7 points worse; low-bar partners keep worse children
Stage 11	without the block, both clock-fixed 100-task children score below the initial prompt; its repeat matches	failed: v1's child -7 $[-13.75, -1.0]$, v2's $+2$ $[0.0, +5.0]$; repeat matches on 100 tasks
Margin rule	descriptive safeguard criterion on complete strata	met at 12 tasks (15/15 arms, no gain to lose); failed at 100

limit, endpoints, analysis code and source hashes. The selectors are ordinary AB, ordinary BA and crossed.

Recovery ledger. All scientific choices preceded the promoted runs. Earlier attempts exposed four operational defects: ceilings too small to finish, a context overflow past the turn budget, an unenforced step limit in one Pi branch, and unrelated processes taking an allocated GPU. Diagnoses read only process state, provider accounting, exception text, token counts and source control flow, never a response, SQL, reward, grade or contrast. Each recovery kept candidates, tasks, cells, estimands and reporting fixed, drew a new order and fresh identifiers, and reran all 800 episodes of the edge. On one machine, after edge $0\to1$ finished, a shared-runtime digest check failed because of concurrent work; a predeclared, outcome-blind closure verified the records and file hashes before the audit.

Integrity disclosure. After S1's cells, endpoints and rules were immutable, a mistaken progress query exposed task identifiers and state-discordance flags for about two partial records, but no response, SQL, grade, candidate text, selector or estimate. An automated layout adapter parsed an outcome-bearing status field before rejecting two missing keys; it created no audit and surfaced no endpoint. Neither event changed a plan or a reporting choice.

S2 and S3. S2 repeats S1 on the same edges and tasks with a harness setting that also shows the trial directory. State a carries the parent's discovery counter and state b the child's later one, with fresh identifiers; only harness-written strings differ from S1. S3 uses this configuration on all three edges and the 400 discovery and calibration tasks, in 20 disjoint shards with two exact repeats (9,600 episodes). The children were selected on these tasks, so S3 estimates η only. Its frozen rule calls η

nonzero if both repetitions' intervals exclude zero with one sign on some edge, and otherwise reports a bound if every half-width is at most 2.8 points.

Stage 2a and Stage 2b. Stage 2a crosses both lineage-2 edges in two runs. Runs differ only in their run directory and fresh identifiers. The frozen question is whether the crossed selector keeps the same candidate in both runs; the run tests of Appendix I were frozen before any lineage-2 outcome. Two runs stopped on context length, the second of $0\to2$ after 337 of 800 episodes and the first of $0\to1$ after 385: requests of 132,853 and 132,419 tokens exceeded the 131,072-token context, and the Stage-2a plans, unlike lineage 1's, had no rule to end such an episode instead of the run. We reran both runs in full under the lineage-1 rule, which scores an over-limit request as a failed episode, with the same candidates, tasks and cells, a new run directory and fresh identifiers. The diagnosis read only status, error text and logs, and the replacements were frozen before any lineage-2 outcome was read. This reverses our earlier frozen note not to recover these runs, and it changes the instrument after the two surviving runs had episodes, which the Stage-2b freeze had ruled out. The policy touched 12 requests in the recovered run of $0\to2$ and none in $0\to1$. Stage 2b, three more runs of $0\to2$, uses the same policy and a per-run provider lock.

The lost regret study. Before any S1 result, we froze a study of all four lineage-1 candidates on 50 calibration tasks under two new equal-token states and two repeats (800 episodes), to score selectors by fresh-task regret, with a zero-call budget analysis. The machine holding it was lost before its audit, and we did not rerun it.

B Closed Mechanism Results

Table 7: Closed interventions. A/A: tasks on which identifier A matches its exact repeat. Other columns count tasks on which identifiers A and B differ; Calls/tokens: tasks whose call or token counts differ; Tokens: total tokens processed; —: not an endpoint. *Post hoc SQL-block parser, 59 tasks. With latency visible, the repeat check passed 30 tasks; with latency removed, the clock task is dropped.

Study	n	A/A	First query	Final SQL	Grade	Calls/tokens	Tokens
Qwen, first turn	240	240/240	48/240	—	4/240	—	3.370M
Episode, latency visible	60	30/60	14/60	16/60	0/60	24/60	1.847M
Episode, latency removed	59	59/59	14/59	18/59	1/59	17/59	1.865M
Mistral, native tools	60	60/60	24/59*	—	1/59*	—	0.597M
Mistral, fenced SQL	100	100/100	43/100	43/100	5/100	—	1.036M

Table 8: First-turn identifier intervention by task cohort: two cohorts of 30 training-split tasks (original and prospective, frozen at different times), reused across the four configurations (native: default context; YaRN: extended context). All four grade changes fall in the original cohort; all A/A repeats match.

	First-query changes		Grade gains/harms	
Model and context	original	prospective	original	prospective
Qwen3-8B, native	6/30	8/30	0/1	0/0
Qwen3-8B, YaRN	3/30	5/30	0/0	0/0
Qwen3-32B, native	6/30	9/30	2/0	0/0
Qwen3-32B, YaRN	7/30	4/30	1/0	0/0

Identical copies and the repeat check. Table 9 details the identical-copy test of Section 2. Only 30 of 60 tasks pass the prespecified complete-trajectory repeat gate. On them, identifier changes the first query on 7 and the final SQL on 10, with no grade change. The 60-task counts in Table 7 describe the runs after diagnosis, not causal estimates.

Mistral. A first Mistral study used the model's native tool-call format on 60 tasks; none of its 180 outputs followed the frozen contract. A post hoc parser that accepts exactly one SQL fence, frozen before any SQL difference was read, admits 59 tasks. On them, identifiers A and B differ in SQL on 24, with one harm. A frozen confirmation on 100 further tasks requires one SQL fence. Its A/A runs agree on all 100 tasks. A/B queries differ on 43, with three grade gains and two harms (paired exact $p = 1$). Its first audit stopped on the grader's wall-clock `seconds` field; before any contrast was opened, a recovery compared only the official status and score.

Table 9: Byte-identical copies: any strict choice is false. The crossed row is reconstructed exactly from the observed cells.

Selector	Strict choices in 5 blocks	Correct outcome
Ordinary, accuracy only	1	tie in all 5
Ordinary, accuracy → calls → tokens	5	tie in all 5
Crossed over both states	0	tie in all 5

C THE QWEN3-32B EDGE

The frozen design crosses both candidates of an independently generated Qwen3-32B GEPA edge, $0{\to}1$, over two token-matched states with two exact repeats on 100 test tasks. An external grader scores the first SQL proposal. The state changed 32 child and 30 parent first responses (31 and 28 first queries; Table 15) and six and four grades. Every cell repeated exactly. Every point selector keeps the child. The crossed contrast, $+6$ points, has a frozen percentile interval of $[+0.8, +12.8]$ but a wild-bootstrap interval of $[-0.25, +13.5]$, so the interval gate keeps the parent. The shared effect η is 1 point ($r = 0.00$). At 10 tasks, ordinary and common-state evaluations disagree on 23 percent of decisions and fresh minibatches on 48 percent (Figure 7). Under state b alone the child's lead excludes zero, $+7$ $[+1.8, +13.9]$; under state a it does not, $+5$ $[-1.7, +12.6]$. The interaction, $+2$ $[-3.2, +7.5]$, is unresolved. The scope is one edge's first SQL proposal.

D HOSTED MODELS

We sent lineage 2's endorsed-edge first requests, byte-identical to those of the Stage-4 arms, to DeepSeek-V4.1-Flash and GLM-5.3-Flash in non-thinking mode. The design covers 100 held-out tasks, parent and child, the directories of arms v1 and v2, two exact repeats, and temperatures 0 and 0.6. It was frozen before any call; the DeepSeek half at temperature 0.6 was added, for cost reasons, before the first half was analyzed. A repeat measures serving or sampling noise, and a directory change adds the state's effect (Figure 6; exact within-unit and 11-cluster sign-flip tests). GLM emitted short reasoning in about a fifth of its responses; we read the answer, as the harness does. The directories are not token-matched here, and their prompts differ by at most 14 tokens. DeepSeek served 47 percent of prompt tokens from its prefix cache, and state pairs diverge at the path block, so part of its excess may reflect a different serving path; we could not stratify by cache status. The frozen prediction that the state adds to repeat noise at temperature 0 held for DeepSeek, with a decision excess of 8.8 points $[+0.7, +19.9]$ at 10 tasks, and failed for GLM.

E SOURCE-LEVEL CHECKS

Two zero-call audits trace evaluator paths in other systems. HSI (Zhou, 2026), revision `97a022b`, passes a `run_<timestamp>/repo` path into its evolution and selection inputs. Its context changes under equal-length A/B paths and equalizes under a stable alias; action timestamps stay out of model context. The pinned Pi adapter for τ^2 tasks (Barres et al., 2026) maps distinct request roots to one alias built from domain, task and trial, and seeds its user simulator from the same values. Such designs keep compared candidates on identical strings; they do not show behavioral invariance.

F DECISION SIZES AND STATE EXPOSURE IN PUBLISHED LOOPS

Table 10 shows how published loops decide.

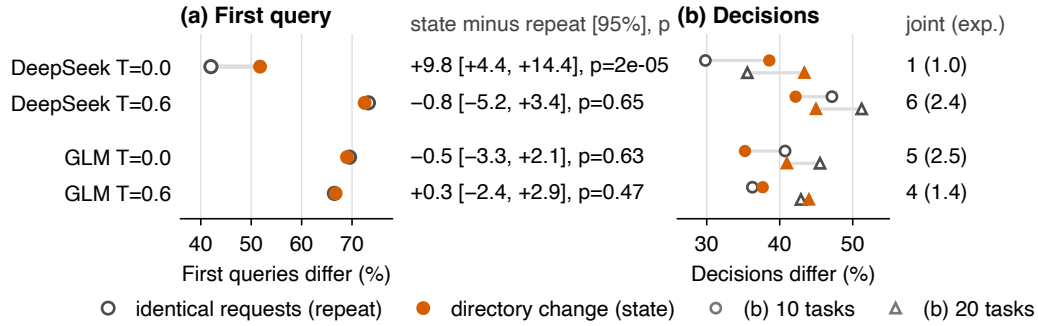


Figure 6: Hosted models, first turns on 100 held-out tasks. Open gray markers: identical requests (repeat noise); orange: a directory change (state). (a) First queries that differ, with the state-minus-repeat excess, its 95% interval and one-sided p. (b) Keep/reject decisions that differ at 10 tasks (circles) and 20 tasks (triangles). Right: joint grade flips of parent and child, observed (expected under independence).

Table 10: How published loops decide, and whether evaluator strings reach the evaluated model, from primary sources (Agrawal et al., 2026; Opsahl-Ong et al., 2024; Zhang et al., 2026; Hu et al., 2025; Robeyns et al., 2025; Yang et al., 2024; Esakkiraja et al., 2026); MIPROv2 defaults from DSPy's source. Only GEPA's pool admission compares a child with its parent on a few tasks; our replay borrows that size, not these loops' exposure.

System	Retained object	Tasks per decision	Evaluator strings in context
GEPA	child enters the pool if it beats its parent on a minibatch; final pick best on the validation set	3 (admission); 45–300 (final)	none on its benchmarks; a container ID in its Terminal-Bench adapter (Table 11); Qwen3-8B at T=0.6
MIPROv2	best instruction and demonstrations under Bayesian search	35 per minibatch	none (fixed inputs)
DGM	compiling agents that can still edit code	10, 60, 200 (screens, not parent comparisons)	none in the instruction; `/testbed/`; T=1.0
ADAS	every generated agent	20 (ARC validation)	round index; random agent IDs
SICA	highest utility (score, cost, time)	$\geq$100 (four benchmark sets)	yes: timestamps, agent IDs, durations, costs
OPRO	best 20 instructions	about 262 (3.5% of GSM8K train)	none
StarHarness	must improve the selection mean	about half the benchmark	not determined

Source audit. We read eight agent harnesses and six self-evolving systems at pinned commits, and Claude Code from its documentation (Table 11). Presence comes from code paths, except for DGM's committed logs and Harbor's golden traces, where the strings appear verbatim. Pi removed the date from its system prompt in July 2026 for prompt caching, and Harbor's tests normalize Terminus container identifiers. File and line references will be released with the code.

G Proofs

Proposition 1 (What one ordinary assignment identifies)**.** *Suppose only $\hat{\Delta}_{ab}$ is observed. (i) Without further restrictions, the finite-sample two-state effect $\hat{\tau}_\times$ can lie anywhere in an interval of width one containing $\hat{\Delta}_{ab}$, however many tasks or reruns there are. (ii) If each candidate's outcome is known to change between the two states on at most f_c tasks, then $|\eta| \leq (f_\mathrm{P} + f_\mathrm{C})/(2|T|)$ and $\hat{\tau}_\times \in \hat{\Delta}_{ab} \pm (f_\mathrm{P} + f_\mathrm{C})/(2|T|)$.*

Table 11: Harness-generated strings in model-visible context, from source at pinned commits. R: random per run; S: set by the date or the run's progress; F: fixed per deployment (`/testbed` and `/app` in every deployment). †Conditional. *Aider's benchmark strips timings. ‡Documentation only; n.d.: not determinable. §Proposer only. Sources: Zechner (2026); OpenHands (2026); SWE-agent (2026b;a); OpenAI (2026); Google (2026); Aider-AI (2026); Harbor Framework (2026); Agrawal et al. (2026); Zhang et al. (2026); Robeyns et al. (2025); Hu et al. (2025); Algorithmic Superintelligence (2026); Sakana AI (2026).

System	Paths/IDs	Date/counters	Latency/cost	Kind
Agent harnesses				
Pi	cwd; temp file name†	date until v0.80.6	–	F, R†
OpenHands	cwd per shell result; conversation UUID†	date and time†	–	F, S, R†
SWE-agent	`/testbed`	–	–	F
mini-SWE-agent	`/testbed`	steps left†	time left†	F, S†, R†
Codex CLI	cwd; chunk ID per shell call	date, time zone	wall time per shell call	F, S, R
Gemini CLI	workspace and temp dirs; process-group ID	date	–	F, S, R
Aider	–	date	–*	S
Terminus	`/app`; container ID	–	–	F, R
Claude Code‡	cwd; memory paths	n.d.	n.d.	F
Self-evolving systems (candidate harness)				
GEPA (Terminus)	`/app`; container ID	–	–	F, R
DGM (own)	`/testbed`; container ID†	–	–	F, R†
SICA (own)	fixed workdir; agent IDs	timestamp; iteration	duration; tokens, cost	F, S, R
ADAS (own)	agent IDs†	round; generation	–	S, R†
OpenEvolve§	temp file path†	attempt index	–	S, R†
ShinkaEvolve§	timestamped run dir†	generation†	–	S†

Proof of Proposition 1. (i) Write the missing cells as $u_t = Y_t(\mathrm{C}, a)$ and $v_t = Y_t(\mathrm{P}, b)$, and the observed ones as $\alpha_t = Y_t(\mathrm{P}, a)$ and $\beta_t = Y_t(\mathrm{C}, b)$. Then $\hat{\tau}_\times = (2|T|)^{-1}\sum_t(u_t - \alpha_t + \beta_t - v_t) = \{\hat{\Delta}_{ab} + |T|^{-1}\sum_t(u_t - v_t)\}/2$. With $u_t, v_t \in [0,1]$ and no other restriction, $|T|^{-1}\sum_t(u_t - v_t)$ ranges over $[-1,1]$, so $\hat{\tau}_\times$ ranges over $[(\hat{\Delta}_{ab} - 1)/2, (\hat{\Delta}_{ab} + 1)/2]$, an interval of width one (Manski, 1990). Reruns reveal α_t and β_t again and leave it unchanged. For binary outcomes the attainable values form a lattice with spacing $1/(2|T|)$ whose convex hull is this interval. (ii) By Equation 3, $\hat{\tau}_\times = \hat{\Delta}_{ab} - \eta$, and $|\eta| \le \{|\mu_{\mathrm{P}b} - \mu_{\mathrm{P}a}| + |\mu_{\mathrm{C}b} - \mu_{\mathrm{C}a}|\}/2 \le (f_\mathrm{P} + f_\mathrm{C})/(2|T|)$, because each candidate's mean moves by at most $1/|T|$ per task whose outcome changes. □

Proposition 2 (Search selects on the facet)**.** *Take one incumbent and K byte-identical challengers, each scored under its own state. (i) If all $K+1$ copies are scored under fresh draws, their scores $(Z_0, \ldots, Z_K)$ are exchangeable, and M copies tie at the maximum, a best-score selector makes a strict false update with probability $1 - \mathbb{E}[M]/(K+1)$, which is $K/(K+1)$ when ties have probability zero. (ii) If instead each retained incumbent keeps the score at which it was selected, scores are i.i.d. and continuous, and each round draws K new challengers, the probability of a false update in round r is $K/(1+rK)$, and the expected number over R rounds is $\sum_{r \le R} K/(1+rK)$, which grows like $\ln R$. (iii) Neither probability changes under exact reruns, provided each candidate–state cell is deterministic under the fixed execution conditions. Under that condition, scoring every copy on the same multiset of states makes identical copies tie.*

Proof of Proposition 2. (i) Let I_i indicate that copy i attains the maximum. Exchangeability gives $\mathbb{E}[I_i] = \mathbb{E}[M]/(K+1)$ for every i, and a strict false update occurs exactly when the incumbent is not a maximizer. (ii) In round r the incumbent's score is the largest of $1 + (r-1)K$ i.i.d. draws; the round's K challengers hold the maximum of all $1 + rK$ draws with probability $K/(1+rK)$ by symmetry. The sum is a harmonic-type series. (iii) Reruns reproduce every Z_i. Identical copies under one multiset of states have identical averages. □

With discrete scores, ties cap the first probability; for a two-point score with high value probability p it is $(1-p)\{1-(1-p)^K\}$. For real candidates, crossing equalizes the states but an argmax still selects on the candidate-by-state interaction.

Decision boundaries. With $\tau = \hat{\tau}_\times$, the two ordinary accuracy contrasts have opposite signs exactly when $(\tau+\eta)(\tau-\eta) < 0$, that is $|\eta| > |\tau|$. Because $\min(\Delta_a, \Delta_b) = \tau - |\iota|/2$, non-regression in every audited state is equivalent to $\tau \geq |\iota|/2$. Both statements are algebra on the audited grid; ties, cost tie-breaks and uncertainty need their own rules.

H Decision Replay

An evaluation accepts the child when its per-task contrasts on a minibatch sum to a positive value. Two evaluations of one pair should share no state, and two audited states per task allow two such evaluations. Under common states, one evaluation uses a for both candidates and the other b; this matches two fresh common-state evaluations for any dependence between the candidates. Under ordinary assignment, AB puts the parent under a and the child under b, and BA the reverse. This matches two fresh ordinary evaluations only if the candidates' state effects are independent, which the small correlations of Section 4.1 suggest but do not establish. A random orientation per task would share both cells on half of the tasks, so we do not use it. We use all 161,700 three-task minibatches of 100 tasks and 200,000 draws otherwise, without replacement; the fresh-minibatch comparator uses the crossed rule. In S2 and S3, AB follows the counters' lineage order. For S1's endorsed edge, a database-cluster bootstrap puts the 10-task rate at 22–44%.

Over Section 4's ten edge runs (Figure 7; Table 13), sharing the state changes 10-task disagreement by -5.1 to $+3.8$ points, $+0.07$ on average. A post hoc bootstrap over the five candidate pairs gives a 95% interval of -0.5 to $+1.6$. The pairs share tasks and, within a lineage, candidates, so this interval describes them, not new lineages. At 3 tasks the median is 10% (range 3–17%). In the 400-task counter study, outside its freeze, reversing the state assignment changes 18–25% of the endorsed edge's decisions at 10–50 tasks and a fresh minibatch 42–45%.

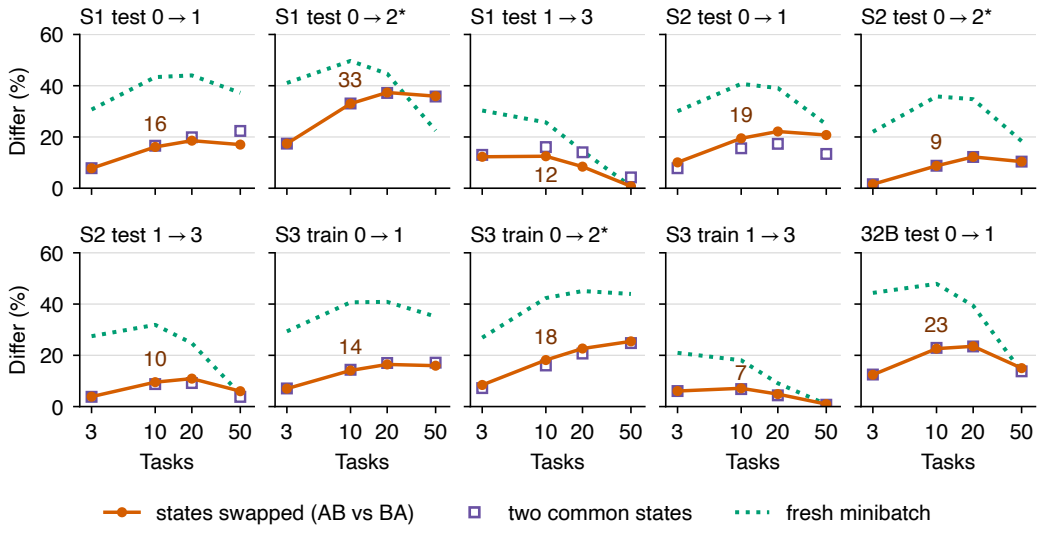


Figure 7: Keep/reject decisions that differ between two evaluations of one minibatch, by tasks per decision: states swapped (AB vs BA), one state shared within each evaluation, and two fresh minibatches (crossed rule). Panel titles give the study (S1–S3: lineage 1; 32B: the Qwen3-32B edge), the task split and the edge. Numbers: AB/BA at 10 tasks; *: endorsed edge. Disagreement rises while more minibatches hold a changed grade, then falls once the contrast is far from zero.

States that are not reused. Lineage 2's frozen regime studies ran edge 0→2 in five runs and 0→1 in two, each run with fresh states and its own directory. Evaluations from different runs therefore

use four distinct states. Post hoc, they disagree at 10 tasks on 13.4% of decisions for 0→2 (run pairs 9.9–16.2%) and 15.2% for 0→1, against 10.6% and 15.9% for the swap within a run; without its recovered run, 0→2 gives 13.8% against 11.3%. Common states give 13.0% and 14.6% across runs, and the frozen analysis had reported 13.0% for 0→2. Reusing states thus does not inflate the replay, and sharing a state lowers the cross-run rate by under one point.

Decision study. With v the within-task state variance of the per-task contrast, the state part of the SD of a mean contrast over n tasks and m draws is $s = \sqrt{v/(nm)}$. Under a normal approximation, two independent evaluations agree with probability $\Phi(d/s)^2 + \Phi(-d/s)^2$ at difference d, which reaches 90% at $d = 1.62\,s$; adding V_{task}/n covers redrawn tasks. Table 12 uses the ten edge runs (v from 0.04 to 0.07), the inputs of Figure 5a. This analysis is post hoc.

Table 12: Post hoc decision study over the ten edge runs. State SD: state part of the SD of a mean contrast, median over edge runs. Last columns: candidate difference (points) for 90% agreement between two independent evaluations, median (range). The approximation ignores ties, which matter most at three tasks (Figure 5a), and treats states as drawn within a run; it leaves the frozen margins unchanged.

Tasks	State draws per task	State SD (points)	Difference for 90% agreement fresh states	fresh states and tasks
12	1	6.6	10.7 (9.3–12.4)	19.4 (16.8–23.7)
12	4	3.3	5.4 (4.7–6.2)	16.9 (13.0–21.2)
50	1	3.2	5.2 (4.6–6.1)	9.5 (8.2–11.6)
50	4	1.6	2.6 (2.3–3.0)	8.3 (6.3–10.4)
100	1	2.3	3.7 (3.2–4.3)	6.7 (5.8–8.2)
100	4	1.1	1.9 (1.6–2.1)	5.8 (4.5–7.3)
300	1	1.3	2.1 (1.9–2.5)	3.9 (3.4–4.7)
300	4	0.7	1.1 (0.9–1.2)	3.4 (2.6–4.2)

I PER-EDGE RESULTS

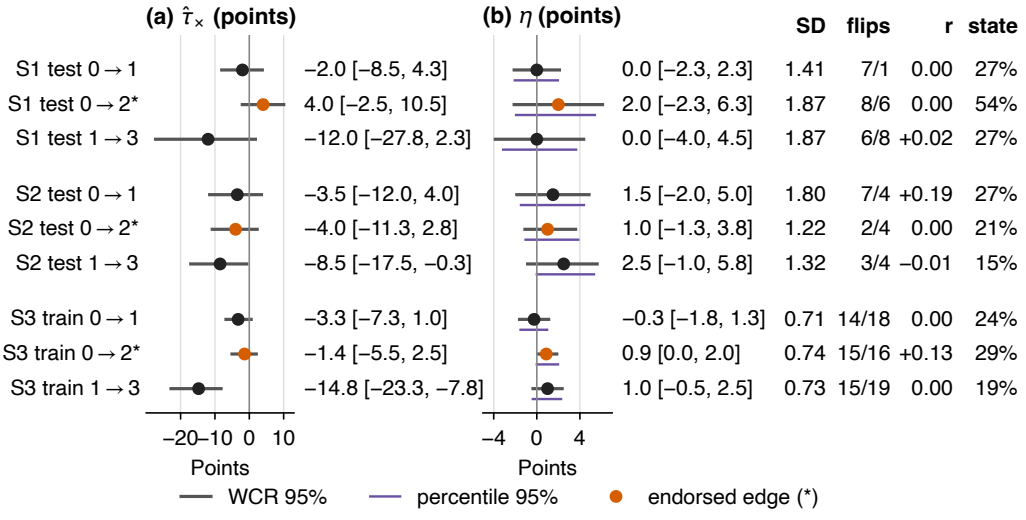


Figure 8: Per-edge facet summary in accuracy points: crossed contrast and average shift η with restricted wild cluster bootstrap-t (WCR) 95% intervals (percentile intervals also for η), randomization SD, grade flips (parent/child), effect correlation and state share of contrast variance.

The frozen bias bound. S3 declared a bound if no edge's η resolved from zero and every percentile half-width stayed within 2.8 points. It resolved as a bound, with percentile upper limits of 1.0, 2.0

and 2.4 points, wild-bootstrap limits of 1.25, 2.00 and 2.50, and a simultaneous band up to 3.0. The two exact repeats differ in one of 4,800 cells, on $1\to3$, so the frozen report gives that edge's η as $+1.0$ to $+1.1$.

Joint flips. With binary grades, a task where one candidate changes carries a shared effect and an interaction; joint changes in the same direction carry only the shared effect, and opposite ones only the interaction. Over S1, S2, S3 and the Qwen3-32B edge, both candidates change on 0, 0, 0, 1, 0, 0, 2, 2, 0 and 0 tasks, against 0.07 to 0.71 expected under independence.

Run-level tests. Table 14 gives three within-task permutation tests: same-run agreement (clustering), a mean shift and a whole-run effect. The first two reject under any task-by-run interaction, which shrinks with more tasks, so neither isolates a persistent run effect. For lineage 1, specified after both studies were read, S1 and S2 serve as two runs; on the endorsed edge the contrast moves from $+4.0$ to -4.0 ($+8.0$, $[-1.75, +17.0]$), and at 50 tasks single-state decisions disagree on 23% of minibatches within a run and 69% across runs. Over lineage 2's five frozen runs of $0\to2$, the between-run SD of the crossed contrast is 2.9 points against 2.1 expected under exchangeable states ($p = 0.10$; whole-run $p = 0.27$), and all five runs keep the parent.

Lineage 2, per run. Table 13 covers all seven lineage-2 edge runs: Stage 2a's two first-attempt and two recovered runs, and Stage 2b's three runs of $0\to2$. Every child trails its parent on held-out tasks. The shared effect ranges from -3.5 to $+4.0$ points; the recovered run of $0\to1$, $+4.0$ $[+0.5, +6.5]$ (randomization $p = 0.035$), is the only identifier-only interval in either lineage that excludes zero. Parent and child change together on 2 of 39 flips. The first-attempt runs repeat 399 of 400 complete trajectories and the others all 400; every run repeats all grades.

Table 13: Lineage 2 edge runs (100 held-out tasks, 11 database clusters); r: recovered run (lineage-1 rule for over-long requests). Effects in points with WCR intervals. Corr.: parent–child correlation of per-task state effects. Flips P/C: grade changes between states. State share: state's share of a single-state contrast's variance. AB/BA, Common a/b: % of decisions that differ between the two evaluations at 10/20/50 tasks.

Run	Crossed $\hat{\tau}_\times$	Shared η	Corr.	Flips P/C	State share	AB/BA	Common a/b
$0\to1$, run 2	-8.0 [-13.8, -1.5]	-1.0 [-3.0, +1.2]	-0.25	3/5	35%	12/9/2	16/14/5
$0\to2$, run 1	-7.5 [-17.2, +2.0]	-1.5 [-5.2, +2.2]	+0.01	7/4	32%	16/13/4	18/17/8
$0\to1$, run 1^r	-7.0 [-14.5, +0.5]	+4.0 [+0.5, +6.5]	+0.19	6/4	27%	19/21/15	14/14/6
$0\to2$, run 2^r	-12.0 [-24.2, -1.0]	+0.0 [-5.2, +5.5]	+0.02	7/3	24%	8/5/0	10/8/1
$0\to2$, run 3	-10.0 [-18.8, -1.2]	+1.0 [-1.8, +3.8]	+0.00	2/8	28%	10/8/1	10/8/1
$0\to2$, run 4	-15.5 [-24.5, -7.2]	+3.5 [-0.2, +8.0]	+0.14	6/7	32%	6/3/0	3/1/0
$0\to2$, run 5	-11.5 [-23.5, -1.2]	-3.5 [-6.8, +0.0]	+0.14	5/8	30%	13/10/1	10/6/0

Table 14: Run-level results. Point, Gate: candidate kept in each run by the crossed point selector and by the interval gate (WCR lower limit above zero); C child, P parent. Excess SD: run-level SD beyond that expected under exchangeable states, $\sqrt{\max(0, V - \mathbb{E}V^*)}$. p-values from within-task permutations (clustering one-sided); whole-run p permutes runs with their two states. Holm-adjusted p in parentheses.

Edge	Crossed contrast by run	Point	Gate	Excess SD	Mean p (Holm)	Cluster p (Holm)	Run p (Holm)
Lineage 1, S1 vs S2 (exploratory)							
$0\to1$	-2.0, -3.5	PP	PP	0.0	0.720 (0.720)	0.0007 (0.0015)	0.796 (0.969)
$0\to2^*$	$+4.0$, -4.0	CP	PP	5.3	0.008 (0.025)	0.0063 (0.0063)	0.051 (0.152)
$1\to3$	-12.0, -8.5	PP	PP	1.2	0.328 (0.655)	0.0005 (0.0015)	0.485 (0.969)
Lineage 2, Stage 2a with recovered runs, two runs (frozen)							
$0\to1$	-7.0, -8.0	PP	PP	0.0	0.827 (0.827)	0.2088 (0.2088)	0.850 (0.850)
$0\to2^*$	-7.5, -12.0	PP	PP	2.5	0.137 (0.273)	0.0088 (0.0176)	0.235 (0.471)
Lineage 2, five runs of $0\to2$ from Stages 2a and 2b (frozen)							
$0\to2^*$	-7.5, -12.0, -10.0, -15.5, -11.5	PPPPP	PPPPP	2.1	0.098 (0.098)	$<$0.0001 ($<$0.0001)	0.265 (0.265)

J PARENT–CHILD INDEPENDENCE AT THE RESPONSE LEVEL

Grade flips are rare, so we also compare response-level changes. For each task, $P = 1$ if the parent's first response, first query or final query (queries whitespace-normalized) differs between states a and b (repeat 1), and $C = 1$ if the child's does (Table 15; Mantel–Haenszel pooling, exact stratified test). S1 and S2, and the two lineage-2 runs of each edge, cross the same candidates under different string pairs. Pairing a candidate's change with the other's under a different pair keeps task fragility but removes shared responses to one string. Over 500 task–edges, matched and mismatched pairings give 63 and 44 joint first-query changes ($p = 0.004$) and 71 and 66 final-query changes ($p = 0.52$); each candidate's own changes recur across strings (odds ratio 2.3 to 3.4). Figure 1c shows S1's endorsed edge: first queries change on 18 and 25 of 100 tasks, final queries on 27 and 27, grades on 8 and 6. Hosted first responses change on 93–99% of calls and carry no information; their first queries co-occur no more than repeat noise. Eight of twenty S3 shards of $1 \to 3$ lack query records, so its query fields cover 240 of 400 tasks.

Table 15: Parent–child co-occurrence of state-induced changes. Parent, Child: share of tasks whose field changes between states. Joint: both change, observed/expected under independence. ϕ: pooled 2×2; p: exact stratified test, repeat 1. Hosted rows pool temperatures 0 and 0.6; the repeat-noise row compares identical requests.

Study	Field	n	Parent	Child	Joint obs/exp	ϕ	p
Qwen3-8B, lineage 1	first query	1640	.198	.215	103/71.4	.12	4×10^{-6}
	final query	1640	.240	.248	126/97.1	.10	1×10^{-4}
	grade	1800	.043	.044	5/3.5	.02	.39
Qwen3-8B, lineage 2	first query	400	.172	.203	17/14.3	.05	.41
	final query	400	.217	.223	21/20.1	.02	.77
	grade	400	.058	.040	2/0.9	.06	.22
Qwen3-32B	first query	100	.280	.310	8/8.7	−.03	.81
Hosted, state	first query	400	.647	.647	191/169.9	.26	4×10^{-6}
Hosted, repeat noise	first query	800	.620	.636	374/327.2	.31	1×10^{-13}
Hosted, state	grade	400	.070	.117	6/3.3	.08	.12

K LOOP EXPERIMENTS

K.1 SETUP AND CONTROLS

Arms and endpoints (Table 16). Each setting has a no-block arm, four exposure arms whose run directories differ only in a version suffix, an A/A arm that shows the first arm's directory string from another physical directory, and three optimizer-seed arms in that directory. E1 and E2 ask whether the no-block and A/A arms repeat their references; for GEPA on Qwen3-8B the no-block arm reruns sealed lineage 2. E3 asks whether exposure arms diverge early and keep at least two distinct candidates. E4 asks whether kept candidates spread more in held-out accuracy than the initial prompt across directories. E5 retests each exposure arm's kept candidate in two fresh directories; no-block arms have no E5, since their source binding differs. Suffixes c and s mark the clock-fixed reruns at 12 and 100 validation tasks (Stages 9a and 10a), where E4s asks only whether the kept prompts differ. OPRO's second no-block arm and its seed arms were canceled when its node failed, so OPRO has no E1 and no seed comparison. Each setting ran on one H200 node, OPRO's on a second node with the same software; its retests ran on the first after hash-verified copying. Every plan and schedule change was frozen, with bound hashes, before the outcomes it concerns; some freezes carry hand-written times.

The exposure. The exposure arms place the Pi runner's two path contexts, an `AGENTS.md` block and the line `Current working directory: <cwd>`, before the instruction; Pi appends them after it. The path encodes the visible run directory, the candidate's hash and the task, so it varies by candidate and task. The block lowers the initial prompt's validation score and raises its held-out accuracy, so we compare exposure arms with each other. Reflection and the meta-prompt never see the system message; the strings reach OPRO only through scores and GEPA also through trajectories. GEPA's Terminal-Bench adapter is a native case (Appendix F).

Controls as frozen. E1 and E2 require every training record to repeat. In the small Qwen configuration both controls repeat every decision and held-out outcome, but their trajectories differ from records 44 and 30, where a query reads SQLite's clock. As frozen, both controls fail, and the rule does not attribute divergence to the strings. The first mismatch is a tool observation in the controls and a model response at record 1 or 2 in every exposure pair. Using that after the failures would be post hoc, so Stage 9a reran the controls with the clock fixed.

Table 16: Loop settings before the clock fix, frozen endpoints; the clock-fixed reruns of Stages 9a and 10a repeat every record (Table 6). E1, E2: repeats of decisions and trajectories. First divergence: median record over exposure-arm pairs, for traces and candidate evaluations. Gain: kept minus initial held-out accuracy in the arm's directory; own/fresh: its mean in own and two fresh directories (E5). Initial: the initial prompt's held-out accuracy across directories. Distinct: kept prompts among exposure arms, and among seeds or directories.

Loop	E1 dec./traj.	E2 dec./traj.	First divergence trace	First divergence eval.	Distinct kept	Gain per arm	Initial held-out	Mean gain own/fresh	Distinct seeds/dirs
GEPA, Qwen3-8B	yes/no	yes/no	1	14	4	-14 to 0	50–53	-9.8 / -7.1	4 / 4
GEPA, Mistral-Small	yes/yes	yes/no	1	20–30	2	-2 to 0	46–50	–	–
OPRO, Qwen3-8B	–	yes/yes	1	13–49	2	-9 to -6	52–54	-7.5 / -5.4	–
GEPA, 100 val. tasks	–	no/no	1	104–125	4	-11 to -3	49–52	-6.3 / -7.5	–

Table 17: Held-out change of the kept prompt (points; 100 test tasks) with restricted wild cluster bootstrap-t 95% intervals over 11 databases. Descriptive, written after the first arms were read.

Loop	Arm	Initial	Kept	Own dir.	Fresh dir. 1	Fresh dir. 2
GEPA, Qwen3-8B	no block	44	45	+1 [−5.8, +8.2]	–	–
	dir v1	51	37	−14 [−22.8, −4.8]	−11 [−16.0, −6.2]	−7 [−14.0, −0.5]
	dir v1, A/A	51	37	−14 [−22.5, −5.0]	–	–
	dir v2	52	38	−14 [−26.0, −3.5]	−8 [−11.5, −4.5]	−9 [−16.8, −0.8]
	dir v3	53	42	−11 [−20.0, −3.0]	−11 [−18.2, −3.2]	−11 [−18.5, −3.2]
	dir v4	50	50	0 (initial kept)	0 (initial kept)	0 (initial kept)
	dir v1, opt. seed 1	51	45	−6 [−15.5, +3.0]	–	–
	dir v1, opt. seed 2	51	44	−7 [−21.2, +6.5]	–	–
	dir v1, opt. seed 3	51	36	−15 [−28.0, −3.2]	–	–

Clock-fixed controls (Stage 9a). Stage 9a, frozen before its runs, reran the four directories and both controls with every SQLite time reference rewritten to one fixed timestamp, in the model's queries, the submitted SQL and the gold SQL. The model never sees the rewrite. Of 500 gold queries, 11 read the clock, and all others return byte-identical results. Both controls then repeat every record (E1c and E2c hold). The directories first differ at record 1 or 2, in a model response, and keep the same outcomes as in Stage 4. The frozen rule therefore attributes the change in the kept candidate to the string.

K.2 The grid and further directories

The crossed grid. The four directories crossed with optimizer seeds 0–3 give 16 cells; seven had been read when the other nine were frozen (Figure 9). Table 17 gives each arm's held-out change with 95% intervals. The bar is 3 of 12 in v1–v3 and 5 of 12 in v4 at every seed, as predicted, since the seed acts after the initial prompt is scored. The cells keep 8 distinct prompts, and v4 keeps the initial prompt. In a descriptive two-way decomposition of the kept prompt's held-out accuracy, the directory, seed and residual mean squares are 107.6, 5.4 and 5.4 (permutation $p = 0.003$ and 1.0). The directory's share, 0.83, rests on v4: it is 0.24 without v4.

Eight more directories (Stage 4e). Stage 4e, planned after the grid was read but before its own runs, ran v5–v12 at seed 0. It predicted that every new kept child is worse held out and that low bars keep more children than high bars. Seven arms keep children, all 3–16 points worse; the eighth keeps the initial prompt on a tie. No bar reaches 5, so the second prediction is unresolved (Figure 11a). These runs predate the clock fix, which left the original directories' kept prompts unchanged; they do not isolate a causal effect of the bar.

Directory (bar)	Opt. seed 0	Opt. seed 1	Opt. seed 2	Opt. seed 3	Held-out, initial → kept
	cell: kept prompt · its validation wins/12 · held-out change (points)				
v1 (3/12)	A · 4/12 · −14	B · 4/12 · −6	C · 5/12 · −7	D · 5/12 · −15	initial 51%
v2 (3/12)	E · 4/12 · −14	E · 4/12 · −14	E · 4/12 · −14	E · 4/12 · −14	initial 52%
v3 (3/12)	F · 4/12 · −11	G · 6/12 · −11	G · 6/12 · −11	G · 6/12 · −11	initial 53%
v4 (5/12)	initial kept	initial kept	initial kept	initial kept	initial 50%
Stage 4e (frozen before outcomes)					
v5 (3/12)	F · 4/12 · −9	not run	not run	not run	48 → 39; rule: initial
v6 (4/12)	initial kept	not run	not run	not run	49 → 49; rule: initial
v7 (4/12)	H · 5/12 · −10	not run	not run	not run	51 → 41; rule: initial
v8 (4/12)	A · 5/12 · −13	not run	not run	not run	51 → 38; rule: initial
v9 (4/12)	A · 5/12 · −16	not run	not run	not run	54 → 38; rule: initial
v10 (2/12)	A · 4/12 · −12	not run	not run	not run	53 → 41; rule: child
v11 (3/12)	A · 4/12 · −15	not run	not run	not run	53 → 38; rule: initial
v12 (2/12)	I · 5/12 · −3	not run	not run	not run	51 → 48; rule: child

Figure 9: Directory × optimizer-seed grid (v1–v4) and the eight Stage-4e directories (v5–v12, seed 0 only; other cells not run). Rows give the directory and its bar. Each cell: kept prompt (letters A–I name distinct children), its validation wins out of 12 and its held-out change in points; darker red marks a larger loss, gray the initial prompt kept. Right: initial held-out accuracy (v1–v4), or initial and kept accuracy and the margin rule's choice (v5–v12).

Seeds without the block (Stage 9b). Without the block, seeds 1–3 keep the seed-0 child, each 1 point better held out, and the bar is 5 of 12.

Directory screens (Stages 9c and 10b). Stage 9c screened 36 directories: bars 2 (11), 3 (15), 4 (9) and 5 (1). The bar-5 directory keeps the initial prompt, and its bar-3 partner keeps a child 11 points worse; the frozen high-bar prediction needed two bar-5 directories and stayed untestable. Stage 10b, frozen before its calls, screened 75 more, v49–v123: bars 2 (19), 3 (41), 4 (13) and 5 (2). v64 (bar 5) keeps the initial prompt, but v86 (bar 5) keeps a child that wins 7 of 12 and scores 46 against 53, so the prediction fails. The low-bar v49 and v50 keep one child, 13 and 12 points worse. Of all 123 screened directories, 119 have a bar below the no-block value of 5 (Figure 10).

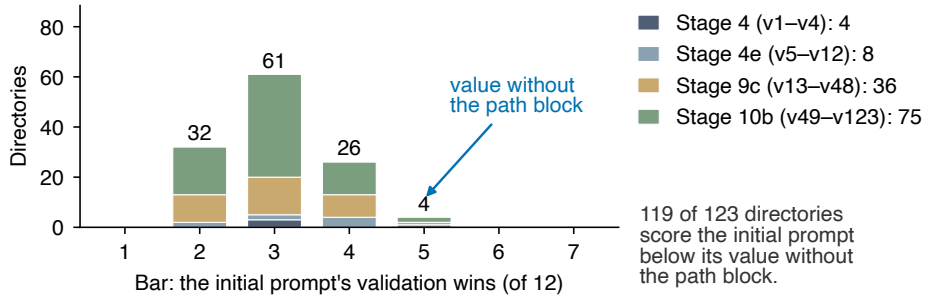


Figure 10: The bar in every screened directory at optimizer seed 0 with the path block; without it the bar is 5.

K.3 A LARGER VALIDATION SET

Stage 6 uses GEPA's reflection minibatch of three, 50 discovery tasks for training and all 100 calibration tasks for validation. The nominal 800 evaluation calls are checked between iterations, so

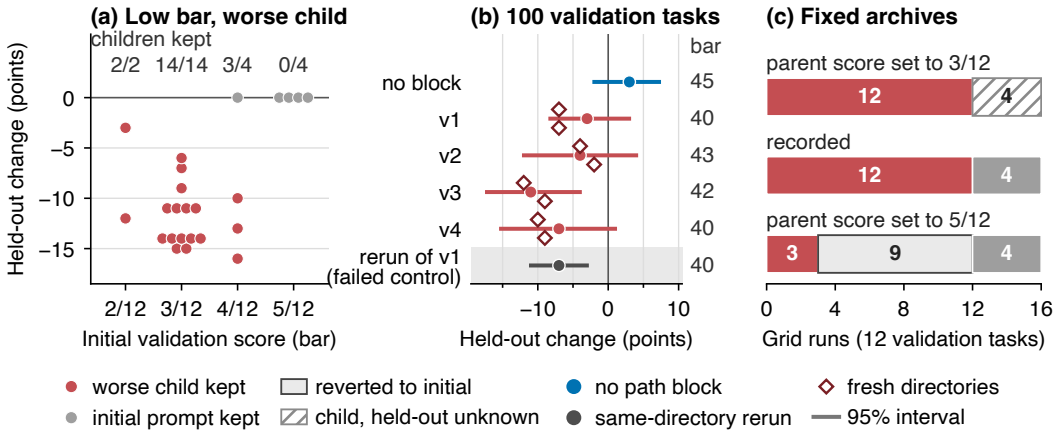


Figure 11: (a) The 16 grid and 8 Stage-4e runs by bar; Figure 4a adds the later screens. (b) Own-directory change and fresh-directory retests at 100 validation tasks. (c) Exploratory replay of the 16 grid archives with only the final parent score replaced.

arms spend 825–895. Decoding stays greedy, unlike published GEPA, so this is an adaptation, not a replication. Two arms first started on a second node and stopped, as designed, when another job took their cards. The study was rerun on the first node with the same run names. The aborted arms were read only for diagnosis, but one arm's initial validation score was exposed; the margin audit discloses this.

Core results. All four directories keep different prompts, but the A/A control fails (Table 18). Its first mismatch is a tool observation at record 134, whereas exposure pairs differ in a model response at record 1 and in candidate evaluations at records 104–125. In the frozen E5 retests the kept children stay 2–12 points below the initial prompt, which scores 52 and 51 (Figure 11b).

Table 18: **Larger-validation core, including the failed control.** Wins and held-out accuracy on 100 tasks; gain with post hoc pointwise 95% intervals. Calls exceed 800 because stopping is checked between iterations.

Arm	Val. initial/kept	Test initial/kept	Gain [95% interval]	Calls
No block	45/46	44/47	$+3$ $[-2.25, +7.5]$	895
v1	40/49	51/48	-3 $[-8.5, +3.25]$	855
v2	43/45	52/48	-4 $[-12.25, +4.25]$	862
v3	42/46	52/41	-11 $[-17.5, -3.75]$	825
v4	40/46	49/42	-7 $[-15.5, +1.25]$	883
v1, A/A	40/41	51/44	-7 $[-11.25, -2.75]$	848

Directory pairs. Over all six pairs of kept-prompt scores (Table 19), a post hoc Holm adjustment gives a smallest p of 0.130, so none rejects at 5%. The calibration of Appendix M does not cover these intervals.

Clock-fixed controls at scale (Stage 10a). Stage 10a, frozen before its calls, reran v1, a same-directory copy and v2 with the clock rewrite. The copy repeats all 813 training records and the test outcomes (E2s holds). v1 and v2 first differ at record 1, in a model response (E3s holds), and keep different children (E4s holds): v1's wins 43 validation tasks against 40 and scores 45 against 51 on the test tasks, and v2's wins 46 against 43 and scores 48 against 52. By the frozen rule, the directory string decides which prompt this loop keeps. Each arm matches its Stage-6 counterpart until record 69, Stage 6's first clock read, and then keeps a different prompt. Stage 11 rescored both children (Appendix K.4).

Table 19: Directory pairs at 100 validation tasks: kept-prompt difference (points), pointwise 95% interval, raw and post hoc Holm p.

Pair	Difference	Pointwise 95% interval	Raw p	Holm p
v2 − v1	0	$[-6.5, +6.25]$	1.000	1.000
v3 − v1	−7	$[-15.25, +1.75]$	0.091	0.284
v3 − v2	−7	$[-14.75, +0.5]$	0.071	0.284
v4 − v1	−6	$[-10.5, -1]$	0.022	0.130
v4 − v2	−6	$[-10.75, -1]$	0.024	0.130
v4 − v3	+1	$[-4.75, +7]$	0.746	1.000

The 12-task validation set (post hoc). Every Stage-6 arm scores the initial prompt on all 100 calibration tasks in its first 100 records, so a zero-call check compares the small loops' first 12 tasks with the rest. Without the block the initial prompt wins 45 of 100 and 5 of the first 12, near the random-subset mean of 5.4. With the block it wins 40–43 of 100 but 2–4 of the first 12, against random-subset means of 4.8–5.2 (7th to 35th percentile). The block thus costs 8–25 points on the first 12 tasks and 1–3 points on the other 88. Low bars reflect the block and this validation set together, the fragility that Section 5.3 sizes.

K.4 Rescoring without the path block

Because the block raises the initial prompt's held-out accuracy, Stage 8 rescored every distinct kept child and the initial prompt on the 100 test tasks with GEPA's own adapter and no block, in three RTX 5090 jobs with hash-identical weights. The freeze predicted that every child scores lower (P1), that the initial prompt's score is identical across jobs (P2) and, descriptively, that it equals the H200 record (P3). P2 held (47 in every job). P3 failed (47 against 44), so we compare within jobs. P1 held for all nine 12-task children and failed for three of five 100-task children (Table 20). The nine share one initial-prompt score, so their intervals are pointwise: two exclude zero, and one remains after a Holm correction over nine (bce074, adjusted $p = 0.03$).

Clock-fixed children (Stage 11). Stage 11, frozen before its calls, rescored the two Stage-10a children on the H200 with the clock fixed and no block, with the initial prompt and an exact repeat in one job. The repeat matched on all 100 tasks and complete trajectories (P2), and the initial prompt scored 44. P1 failed: the v1 child scores 37, 7 points lower, with interval $[-13.75, -1.0]$, and the v2 child scores 46, 2 points higher, with $[0.0, +5.0]$. Under the block both had trailed, by 6 and 4 points.

Table 20: Kept children rescored without the path block, 100 test tasks, one RTX 5090 job per group: change against the initial prompt in the same job, with 95% intervals and tasks won or lost.

Val. tasks	Kept child	Child	Initial	Change	95% interval	Better/worse
12	bce074	38	47	−9	[−12.75, −4.75]	6/15
12	6f507d	40	47	−7	[−11.00, −2.75]	9/16
12	1c95fd	42	47	−5	[−14.00, +2.75]	11/16
12	65729c	42	47	−5	[−13.25, +2.25]	8/13
12	624e3d	43	47	−4	[−10.75, +3.00]	7/11
12	d6f745	43	47	−4	[−13.25, +5.75]	7/11
12	f209db	43	47	−4	[−11.50, +3.00]	14/18
12	3d8f51	45	47	−2	[−11.25, +5.75]	7/9
12	6e6165	46	47	−1	[−6.50, +4.50]	7/8
100	527d8c	46	47	−1	[−5.75, +4.00]	7/8
100	82506f (rerun)	46	47	−1	[−9.75, +6.50]	5/6
100	943065	47	47	0	[−5.50, +6.00]	6/6
100	8f9f53	48	47	+1	[−6.50, +9.25]	7/6
100	eac675	50	47	+3	[−9.50, +15.25]	10/7

K.5 Other optimizers and models

OPRO. Our OPRO-style loop (Yang et al., 2024) shows the task, up to eight best instructions with their scores and three discovery questions without answers, and asks for one greedy instruction per step instead of eight samples. Each proposal is scored on the 12 validation tasks; 96 evaluation calls give the initial prompt and seven proposals. The highest score is kept, and ties keep the earliest. Its 8B optimizer suits OPRO less well (Zhang et al., 2024).

OPRO results. The A/A arm repeats every record. The block lowers the bar to 3 of 12 in v1, v2 and v4 and to 4 in v3, against 5 without it (Figure 12). The bar-3 directories first propose the instruction that the no-block loop keeps, which scores 7 of 12 without the block but 4 or 5 with it. Every copy keeps an instruction 6 to 9 points worse held out, while the no-block loop gains 6, so OPRO's large contrast is between harnesses. In two fresh directories (E5; Table 16) the kept instructions stay 4 to 8 points below the initial prompt.

Mistral results. With Mistral-Small-24B, the no-block arm reruns its sealed lineage record for record, and the A/A arm repeats every decision, though its trajectory differs from record 83 on a tool observation. The bar stays at 7 or 8 of 12, against 7 without the block, so few children pass. The four directories first differ at record 1 or 2 and keep the initial prompt in v2–v4 and, in v1, a child 2 points worse (Figure 13).

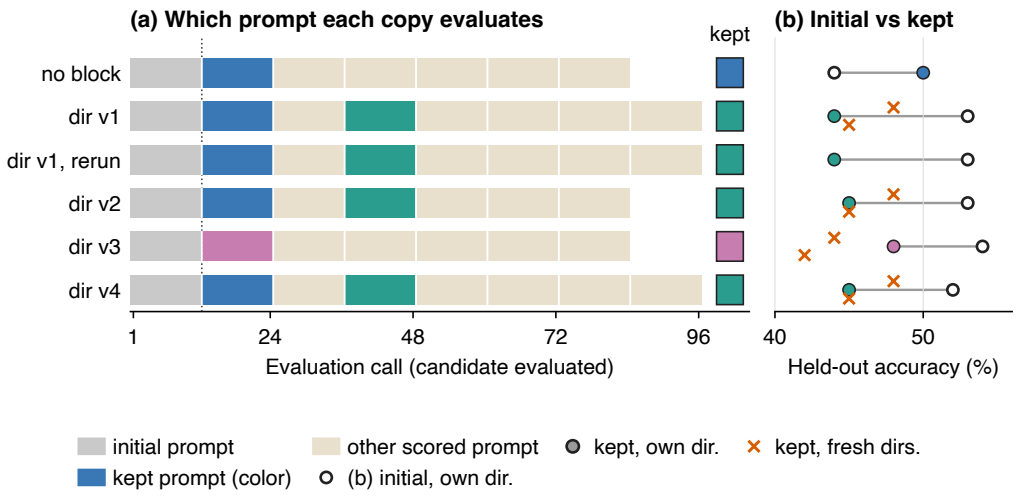


Figure 12: **The OPRO-style loop in detail.** (a) The instruction each copy scores at each evaluation call; colors mark kept prompts, squares the final choice, and the dotted line ends the initial evaluation. (b) Held-out accuracy in the own and two fresh directories.

K.6 The margin rule

Replayed and prospective. The margin rule of Section 5.3 keeps a child only if it wins more validation tasks than the initial prompt by a set margin, taken from the decision study's 90-percent gaps: two tasks at 12 and four at 100. Final choices do not feed back into search, so the replay is exact. All but one of the 16 grid cells and the OPRO arms were read before the rule was chosen. On them it reverts 7 of 12 harmful GEPA choices and none of 4 OPRO choices, and gives up the no-block arm's 1-point gain. We then froze the rule for unread arms, with an addendum written before their completed outcomes were read. At 12 validation tasks, fifteen arms (seven Mistral, eight new directories) all completed. The margin prevents seven of nine regressions, including Mistral's kept child, and sums to $+67$ held-out points, so it meets the frozen criterion. No arm had a gain to lose, so this does not show that the margin preserves improvements. A/A controls and previously exposed seed arms are excluded; a reporting repair enforced these exclusions without changing the

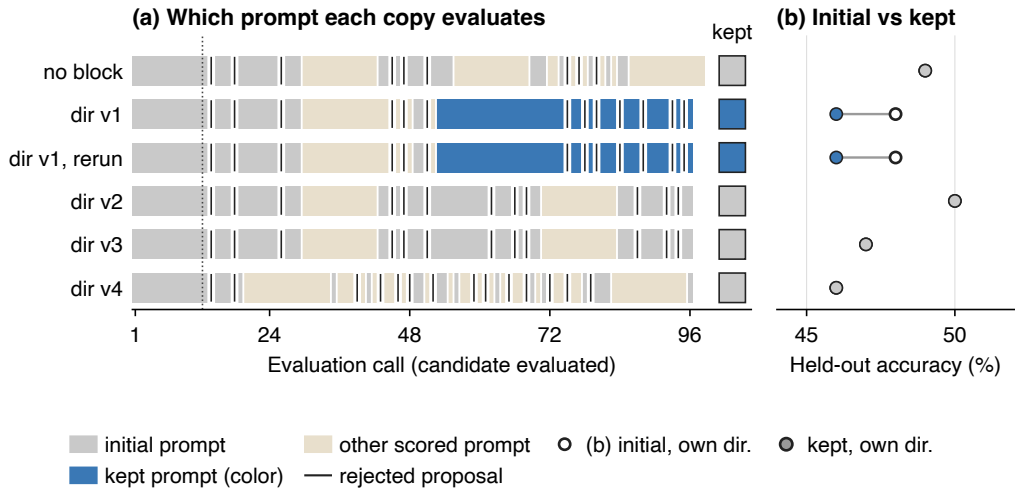


Figure 13: **The Mistral-Small-24B loop in detail.** Panels as in Figure 12; the dotted line ends the initial evaluation.

rule. At 100 validation tasks the four-task margin reverts the no-block gain of 3 points and one of four regressions, summing to one point over five correlated arms. This fails the criterion, which requires more regressions prevented than gains given up and a positive sum, and the failed A/A control limits attribution.

L Fixed-Archive Selector Audit

Scope and reconstruction. This exploratory, zero-call analysis was specified after the sixteen directory–optimizer-seed outcomes were known. It rebuilds each cell's full-validation scores from episode records and reproduces the original final choice. Only the initial prompt's score in the final selector changes, to one of its two observed values, 3/12 or 5/12. Archived children and their scores stay fixed, no child is removed, and ties keep the earliest candidate (Table 21).

Table 21: Final-selector intervention on the sixteen fixed archives. Known regressions: kept candidates already evaluated in their run; a newly kept candidate without that evaluation counts as unknown, not harmful.

Parent score	Choices changed	Children kept	Known regressions	Unknown outcome
3/12	4	16	12	4
5/12	9	3	3	0

The high-bar replay reverts nine of the twelve harmful choices; the other three children score 6/12 and still beat 5/12. This measures selector sensitivity on finite archives, not the share of harm caused during search.

Matched repeats. The same child occurs in the second and fourth directory archives at optimizer seed 0. Its validation wins rise from four to five and its parent's from three to five, so it is kept against parent score three but not against five. This matched example of relative scoring is not a randomized intervention. Fourteen candidate–directory groups have repeated validation occurrences; their 67 pairwise comparisons contain 804 paired task records, all identical in trajectory and grade, though they reuse candidates and tasks. Stage 9a's clock-fixed controls settle full-training repetition (Appendix K.1).

Unreleased follow-up. A full crossing of the fourteen validated candidates over four directories (1,344 episodes) was specified in advance but missed its release condition; it contributes nothing here.

M Evaluation Budgets and Interval Calibration

Inputs and estimand. The ten post hoc inputs are lineage 1's three identifier-only edges and lineage 2's seven identifier-only runs; they share tasks. Each task's empirical state distribution puts equal mass on its two observed states. Regret is the oracle's improvement on this finite reference minus the retained candidate's expected improvement, not new test performance.

Equal-cost allocations. A candidate-pair evaluation costs two episodes, so a budget B buys B tasks once, $B/2$ tasks twice, $B/4$ tasks four times, or $B/2$ tasks crossed over both states (Table 22). At both budgets, one draw per task never gives higher regret, as the variance formula of Section 5.3 predicts. Fixed-state policies stay in the aggregate but do not estimate Q. That formula assumes independent tasks and per-task states; a directory shared by a whole evaluation adds $\mathrm{Var}_s(\mathbb{E}_t[D_t \mid s])$, which more tasks cannot remove and the replay does not estimate.

Table 22: Finite-reference regret (points) for the ten inputs at two pair budgets; state draws are common to the pair within a task. L1:02 is S1's edge 0→2; L2:edge02:r1 is lineage 2's edge 0→2, run 1 of Table 13.

	Pair budget 48				Pair budget 100			
Input	Once	Twice	Four	Crossed	Once	Twice	Four	Crossed
L1:01	0.479	0.643	0.682	0.639	0.070	0.524	0.689	0.495
L1:02	0.955	1.272	1.591	1.203	0.115	0.664	1.089	0.515
L1:13	0.116	0.824	1.850	0.706	0.000	0.080	0.808	0.043
L2:edge01:r1	0.190	0.762	1.264	0.698	0.000	0.171	0.802	0.115
L2:edge01:r2	0.132	0.624	1.179	0.527	0.000	0.100	0.640	0.048
L2:edge02:r1	0.234	0.855	1.470	0.754	0.000	0.197	0.881	0.123
L2:edge02:r2	0.043	0.562	1.406	0.494	0.000	0.030	0.560	0.016
L2:edge02:r3	0.082	0.642	1.377	0.559	0.000	0.061	0.649	0.033
L2:edge02:r4	0.001	0.101	0.588	0.075	0.000	0.000	0.093	0.000
L2:edge02:r5	0.040	0.480	1.279	0.383	0.000	0.025	0.469	0.010

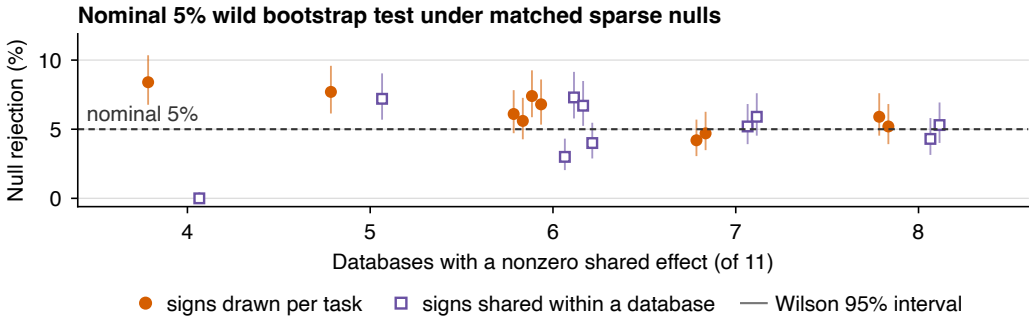


Figure 14: Null rejection rate of the nominal 5% wild bootstrap test in 20 matched sparse-null scenarios (1,000 simulations each), by databases with nonzero effects; Wilson 95% intervals.

Matched null calibration. For each input, the audit keeps the shared-effect scores' magnitudes, nonzero positions and database groups. It draws 1,000 zero-mean sign patterns, with task-level or database-level signs, and tests each with 9,999 Webb draws. Three real-data tests match the prior implementation exactly. The largest rejection rate in Figure 14 is 8.4% (Wilson interval 6.84–10.28%), or 91.6% zero-point coverage. The calibration covers this zero test only. Apart from the five-pair bootstrap of Section 4.2 and the Wilson interval above, every interval resamples tasks by database, never lineages, so each describes the candidates studied.