WHEN SUCCESS IS A LOSSY RECORD: MATCHED-SUCCESS CONTINUATION EVALUATION FOR PERSISTENT AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Agent benchmarks reduce a completed episode to a verdict such as PASS, discarding the environment that later tasks inherit. We formalize this compression failure as **continuation insufficiency**: the excess Bayes risk of predicting future outcomes from the verdict rather than the state under declared continuation conditions. Matched-success evaluation forks equally accepted states into identical frozen continuations and separates transition, preservation, downstream utility, and cost. Across 360 controlled coding pairs, later success differs by 27.50 percentage points; a complementary 240-pair enterprise study shows an 8.75-point shift in the same direction. Among 64 naturally produced successful states, a 2.73-point reference-free separation survives crossed randomization ($p = 0.0023$). On 144 held-out executions in 18 repositories, retaining continuation information avoids 13.19 points of verdict-only choice regret and recovers the historical-state baseline. Replay and recovery identify history compatibility as the principal omitted variable, while a common-snapshot control isolates downstream divergence in Jsonpickle. Success is therefore a lossy record of an agent's work: our protocol detects the missing information, locates its mechanism, and prices the cost of discarding it.

1 INTRODUCTION

A persistent agent produces two outputs, whether or not the benchmark records both: a completed task and the environment state that later tasks inherit. A coding agent leaves a repository; an enterprise agent leaves records and permissions; a memory agent leaves claims that later decisions may reuse. We study this inherited-state problem through executable software states and controlled enterprise workspaces. Many current evaluations verify the first output and compress the second into a verdict such as PASS.[1]

The consequence is easy to state without notation: two agents can both pass the current test while leaving starting points that make the next task easy, expensive, or impossible. A benchmark that stores only the verdict cannot tell those cases apart, so a deployment claim built on that record rests on an untested assumption. The assumption is a measurement-validity condition rather than a style preference. Written formally, a verdict is a state abstraction: every terminal state assigned record z is interchangeable to an evaluator that stores only z. The abstraction is valid for continued interaction when the concrete state adds no predictive information about future value once the continuation context is fixed. We call violations *continuation insufficiency*, written $\mathcal{I}_2(z; \mu_z, \nu)$: a post-success form of underspecification (D'Amour et al., 2022), and a property of the record rather than a quality score for states.

The evaluations surveyed here do not isolate this quantity. Episodic benchmarks reset or discard the state (Liu et al., 2024; Ma et al., 2024; Trivedi et al., 2024; Lu et al., 2025; Yao et al., 2025); long-horizon and software-evolution benchmarks observe one trajectory in which state, future difficulty, executor, and transition compatibility move together (Jimenez et al., 2024; Le et al., 2025; Deng et al., 2026; Chen et al., 2026; Shen et al., 2026; Orlanski et al., 2026; Kapoor et al., 2026). MaintainCoder

[1] Code, frozen protocols, raw outcome rows, and analysis scripts: `https://anonymous.4open.science/r/matched-success-continuation-evaluation-B117/` (anonymous; see the Reproducibility Statement).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

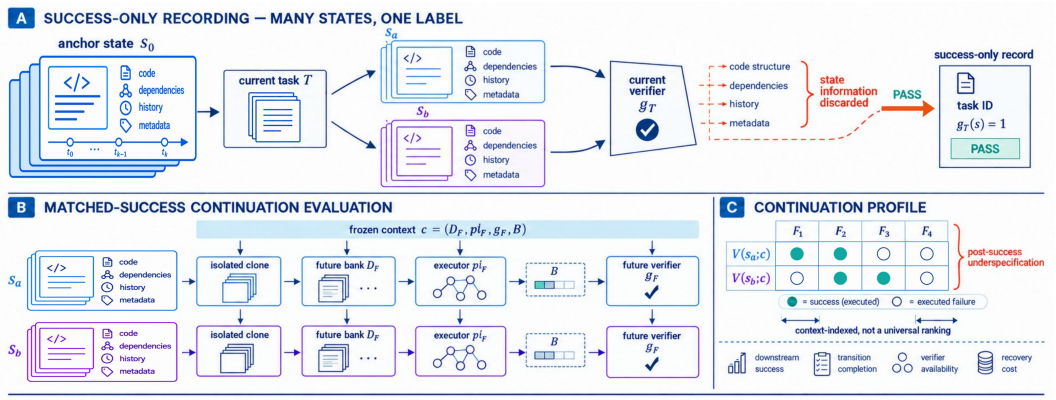

Figure 1: **The same success label can hide different continuations.** Two accepted terminal states collapse through the current verifier to the same stored PASS, which discards state identity. Matched-success evaluation reopens those states under one frozen context $c = (\mathcal{D}_F, \pi_F, g_F, B)$ and compares their future-task profiles. Label mapping: the brace's "post-success underspecification" is continuation insufficiency, and the displayed profile V collects the expected outcomes $v(s, c)$ of Section 2. Colors identify states, not quality; profiles are schematic.

tests maintainability under changing requirements (Wang et al., 2025), and test-based verdicts are known to accept incorrect patches (Wang et al., 2026); we ask whether *correctly* accepted states remain equally usable. Foresight Residual RL studies locally successful terminal states over the remainder of a fixed manipulation chain (Liu et al., 2026); recoverability asks whether one delayed objective remains reachable within an edit budget (Anonymous Authors, 2026); other work values attainable options (Krakovna et al., 2018; Turner et al., 2020; Krakovna et al., 2020) or follows one accumulating history (Shen et al., 2026). Future-task value is therefore not new; the missing object is a validity test for the *record*: condition on a shared accepted verdict, assign identical continuations to distinct concrete states, and determine whether the record preserved the information those continuations need. Our framework supplies that test while separating transition, preservation, utility, and cost.

The missing move is to separate state from continuation context experimentally (Figure 1). On an executable repository substrate, matched-success evaluation forks accepted states, assigns the same frozen context $c = (\mathcal{D}_F, \pi_F, g_F, B)$, and compares outcomes; assigning rather than observing contexts enforces state–context independence.

The empirical argument is layered: controlled pairs and natural states establish that a shared record hides reproducible continuation differences; an independent frozen 18-repository panel prices the record against a verdict-only chooser; and separate eight-repository recovery and five-repository common-snapshot panels locate the price in transition compatibility or downstream utility. Inference is clustered at the model, state, family, or repository level appropriate to each panel.

These findings support three contributions.

- **What should be measured.** A criterion for deciding whether a success record preserves the state information later tasks need: continuation insufficiency, the excess Bayes risk of the record under declared state and context distributions.
- **How it can be tested.** An executable protocol that compares equally successful states under identical continuations, with directed and reference-free estimands, finite-panel regret, and preservation bounds when transitions or verifiers fail.
- **What the experiments reveal.** The stored verdict omits reproducible, decision-relevant information. On the frozen decision panel, continuation evidence recovers the historical-state baseline and quantifies the price of verdict-only choice; replay and recovery trace that price to history compatibility, while a common snapshot provides a downstream mechanism witness without replay.

2 SUCCESS RECORDS AND CONTINUATION SUFFICIENCY

A running example. Two agents resolve the same Jsonpickle issue; both pass the frozen current-task verifier, so the benchmark stores one PASS for each. Which state is better depends on what comes next. Replaying the maintainer's later history favors the historical state: on the alternative, replay stops at a source conflict in `jsonpickle/pickler.py`, so a later task cannot start without repair. From a common snapshot the order reverses. The alternative exposes a reusable reference primitive for plain dictionaries, and a later task that must preserve identity for a dictionary subclass and its `__dict__` is solved only by trajectories that compose through it (Figure 4). One record, two continuations, opposite preferences. This section names the quantity the record fails to preserve; Section 3 makes it measurable.

2.1 RECORDS, FIBERS, AND CONTEXT-INDEXED VALUE

Let $s = \Phi(s_0, T, \xi)$ be the terminal state produced by trajectory ξ on current task T. A benchmark retains only $z = e_T(s)$ for an observation map $e_T : \mathcal{S} \to \mathcal{Z}$; binary success is the case $e_T = g_T \in \{0, 1\}$. The record induces the *success fiber* $\mathcal{E}_z = \{s : e_T(s) = z\}$, the runs a leaderboard has already declared equivalent. Both Jsonpickle states lie in one fiber.

Continuation is defined relative to two distributions that an evaluation must declare. Let μ_z be the distribution of states produced inside $\mathcal{E}_z$, and let ν be a distribution over continuation contexts $c = (\mathcal{D}_F, \pi_F, g_F, B)$: a future-task distribution, an executor (called a consumer when an agent carries out the future task), a verifier, and a budget. For a future outcome G_F, the context-indexed continuation value is $v(s, c) = \mathbb{E}[G_F \mid s, c]$. Value therefore belongs to a state–context pair and never to a state alone.

2.2 CONTINUATION INSUFFICIENCY AND MATCHED WITNESSES

The information a record loses is the extra prediction error paid for seeing z instead of s. Under squared loss and the design measure $\mu_z \times \nu$,

$$\mathcal{I}_2(z; \mu_z, \nu) = \inf_h \mathbb{E}[(G_F - h(z, C))^2] - \inf_f \mathbb{E}[(G_F - f(S, C))^2] = \mathbb{E}_{C\sim\nu}[\mathrm{Var}_{S\sim\mu_z}\, v(S, C)]\,. \quad (1)$$

The record is lossy exactly when equally labelled states still disagree about the future, averaged over the futures the evaluation declares. The optimal predictors are $h^*(z, c) = \mathbb{E}[G_F \mid z, c]$ and $f^*(s, c) = v(s, c)$; we call the record ϵ-continuation-sufficient when $\mathcal{I}_2 \le \epsilon^2$. This is predictive mean-sufficiency under squared loss, not statistical sufficiency for the full law of G_F (Appendix A).

Insufficiency becomes checkable through a two-state witness. For an equally weighted fiber $\{s_a, s_b\}$ with $d(c) = v(s_a, c) - v(s_b, c)$, Equation 1 gives

$$\mathcal{I}_2 = \tfrac{1}{4}\mathbb{E}_\nu[d(C)^2] \;\ge\; \tfrac{1}{4}\big(\mathbb{E}_\nu[d(C)]\big)^2. \quad (2)$$

One matched pair with a nonzero average gap therefore proves that the record loses information; a null mean does not certify sufficiency, since context-specific effects can cancel.

2.3 DECISION CONSEQUENCE WITHOUT A UNIVERSAL RANKING

Lost information matters operationally when it changes a choice. A selector that must commit to one state before the context is revealed has minimum regret, relative to a context-aware oracle,

$$\mathcal{R}^*_{\mathrm{cf}} = \min\{\mathbb{E}_\nu[d_+(C)],\ \mathbb{E}_\nu[(-d(C))_+]\}\,, \qquad d_+(c) = \max\{d(c), 0\}, \quad (3)$$

which is strictly positive exactly when the preferred state reverses inside the support of ν. A selector that sees only the shared record is permutation-invariant, chooses each state with probability $1/2$, and pays $\frac{1}{2}\mathbb{E}_\nu|d(C)|$. Any frozen rule's improvement over uniform choice therefore measures the *record-only regret* it avoids on that panel, not a context-free state score. Transferable state effects and state–context interaction are conceptually distinct, and only the second can reverse a ranking; Appendix A gives the decomposition we do not estimate here.

3 MATCHED-SUCCESS CONTINUATION EVALUATION

3.1 IDENTIFICATION BY THREE MATCHES

Three matches address the standard alternatives. Matching the *current record* removes present-task quality differences because every state passed the same verifier. Matching the *continuation context* removes future and executor differences because every state receives the same task, consumer, verifier, and budget. Matching the *execution condition* controls history compatibility with a shared snapshot or exposes it through a reported transition. Assigned contexts enforce state–context independence under $\mu_z \times \nu$.

3.2 DIRECTED AND REFERENCE-FREE ESTIMANDS

When a reference state is defensible, directed pairs give a signed witness. With fiber i holding (s_i^a, s_i^b) and weights $w_{ik} = \omega_i q_{ik}$ summing to one,

$$\widehat{\Delta}_\nu = \sum_{i,k} w_{ik}\big(Y_{ik}^a - Y_{ik}^b\big), \qquad \overline{\mathcal{I}}_2 := \sum_i \omega_i \mathcal{I}_2^{(i)} \;\geq\; \tfrac{1}{4}\Big(\sum_i \omega_i \bar{d}_i\Big)^2, \tag{4}$$

where $\bar{d}_i = \mathbb{E}_\nu[d_i(C)]$. An observed average paired gap lower-bounds the average risk the record hides, and resampling follows the units that generated states and futures.

Naturally produced states admit no privileged reference, so they are compared crossed rather than directed. There, $\widehat{A}_{\text{state}}$ is the state-count-weighted mean pairwise absolute gap between continuation profiles, and $\widehat{\Psi} = \widehat{A}_{\text{state}} - \mathbb{E}_{\sigma\sim\mathcal{P}_0}[\widehat{A}_\sigma]$ compares it with permutations within family, executor, and future strata. This interpretable L^1 separation is the *primary* natural-state endpoint; the separately reported noise-corrected variance is the confirmatory quantity aligned with $\mathcal{I}_2$.

3.3 FINITE DECISIONS AND THE FAILURE-STAGE DECOMPOSITION

A continuation can fail before the future task ever runs, so outcomes are recorded by stage. Let M indicate reaching the future base, E that the transported current-task verifier executes, Y its verdict, and H the latent event that the earlier accepted behavior is preserved. If an executed pass implies $H = 1$ and an executed failure implies $H = 0$, then

$$L_s(B) = \Pr(MEY = 1) \;\leq\; \Pr(MH = 1) \;\leq\; P_s(B) - \Pr(ME(1-Y) = 1) = U_s(B), \tag{5}$$

with $P_s(B) = \Pr(M = 1)$. This *failure-stage decomposition* keeps "we could not get there", "we could not check", and "it is broken" apart; $U_s - L_s$ is an identified set created by verifier availability, not a confidence interval. Writing $Q_s(c, B) = \mathbb{E}[MHG_F]$ for downstream utility and K_s for recovery cost, $V_\lambda(s; c, B) = Q_s - \lambda K_s$; we fix $\lambda = 0$ and report K separately, because any nonzero exchange rate is a deployment choice.

3.4 EXECUTABLE PROTOCOL

Algorithm 1 states the protocol as six operations: freeze the evaluation before outcomes exist, collect states sharing the record, clone or explicitly transition each one, execute matched continuations while logging the failure stage, estimate separation on development contexts, and only then open heldout outcomes to compare frozen policies. We instantiate it as an evidence ladder in which each tier removes a different alternative explanation.

4 EXPERIMENTAL DESIGN

4.1 RESEARCH QUESTIONS

RQ1 (Existence). Can states sharing one success record produce reproducibly different continuation outcomes? **RQ2 (Decision).** What does storing only the verdict cost a heldout choice, and how does continuation-aware choice compare with state-aware defaults? **RQ3 (Mechanism).** How much of any observed value is transition and history compatibility rather than downstream utility? Cheaper proxies are treated as an alternative-explanation audit for RQ3, not as a fourth question.

Algorithm 1 Matched-success continuation evaluation

Require: accepted-state fiber $\mathcal{S}_z$; development and heldout contexts $\mathcal{C}_{\mathrm{dev}}, \mathcal{C}_{\mathrm{test}}$; selector h; record-aware baselines $\mathcal{B}$

1: Freeze futures, consumers, verifiers, budgets, repeats, h, and $\mathcal{B}$ before any continuation is executed
2: Collect or construct states $\mathcal{S}_z$ that all pass the same current-task verifier
3: **for all** $s \in \mathcal{S}_z$ and $c \in \mathcal{C}_{\mathrm{dev}} \cup \mathcal{C}_{\mathrm{test}}$ **do**
4: Clone s at the declared base, or replay its frozen transition and record the transition outcome
5: Execute the continuation; log M, E, Y, future reward G_F, and cost K, and store the certified value $\widehat{V}_\lambda = MEY\, G_F - \lambda K$
6: Estimate within-record separation and development profiles; choose $s^* = h(\{\widehat{V}(s; c) : c \in \mathcal{C}_{\mathrm{dev}}\})$
7: Open heldout outcomes; compare s^* against uniform, shortest, static-metric, and historical-state policies in $\mathcal{B}$
8: **return** separation, preservation bounds, finite-panel regret, and failure-stage strata

4.2 An evidence ladder of populations, contexts, comparators, and inference

The design is a five-tier evidence ladder mapped onto RQ1–RQ3: controlled pairs and natural states answer RQ1, frozen selection answers RQ2, and replay, recovery, and common-snapshot execution answer RQ3. These are linked tests, not one expanding cohort: the 18-repository decision panel is independent of the eight-repository recovery panel, and the five common-snapshot repositories are a subset of the recovery panel. Each tier exposes accepted states to matched continuations and reports its own denominator rather than pooling heterogeneous executions into one nominal sample size.

Evaluation scale and units. *Controlled pairs:* 600 matched pairs—360 coding and 240 enterprise—yield 1,200 continuation outcomes under five executors. *Natural states:* 90 producer attempts yield 78 successful instances and 64 unique source states; three consumers and four futures produce 936 raw rollouts, aggregated to 768 source-hash–consumer–future cells before state-level weighting. *Frozen decisions:* 18 repositories $\times$ two states $\times$ two heldout futures $\times$ two repeats give 144 executions. *Recovery:* 598 evaluations and 46 complete state–future arms are retained per consumer (1,196 evaluations total) over $B \in \{0, 1, 3, 6, 10\}$; a distinct consumer-matched complete-case control at $B = 10$ contains 428 canonical keys per consumer (856 rows). *Common snapshots:* two consumers contribute 45 sessions each across five repositories (90 total). The prospective construction additionally yields 41 gated repository states, of which 35 enter sequential replay; source conflicts are recorded as transition outcomes rather than silently excluded.

Each later tier removes a specific alternative explanation (Appendix Table 3). Inference resamples the units that generated the data: model and family for controlled contrasts, state within fixed family–consumer–future strata for natural contrasts, and repository after averaging repeats for real-repository panels. The six natural-state family follow-ups are Holm-adjusted, while tier-level tests target distinct estimands. The independent 18-repository decision panel was frozen after the earlier cross-family study; Appendix D reports that study and its prespecified gate. Full denominators, construction studies, protocols, prompts, restarts, and exclusions appear in Appendices B, D, G, and C.

5 Results

5.1 Existence: a shared success record hides reproducible continuation differences

States that pass the same verifier do not continue alike. Across 360 controlled coding pairs, continuation success is 68.33% from extensible states and 40.83% from narrower states, a **27.50 percentage-point gap** with hierarchical 95% confidence interval **[16.11, 40.01]** (Figure 2a). Pooled over both arms, the paired composition calibrates the effect: of 600 matched pairs, 191 continue from both states and 221 from neither, while 154 discordant pairs favor the extensible state and 34 favor the narrower one (Figure 2b). Through Equation 2, this contrast provides a lower-bound witness of positive $\mathcal{I}_2$ for the designed state distribution.

The separation survives every resampling scheme, with the estimator of Equation 4 recomputed under each. A stricter three-way crossed bootstrap gives [7.38, 51.01], with 0.34% of crossed-bootstrap

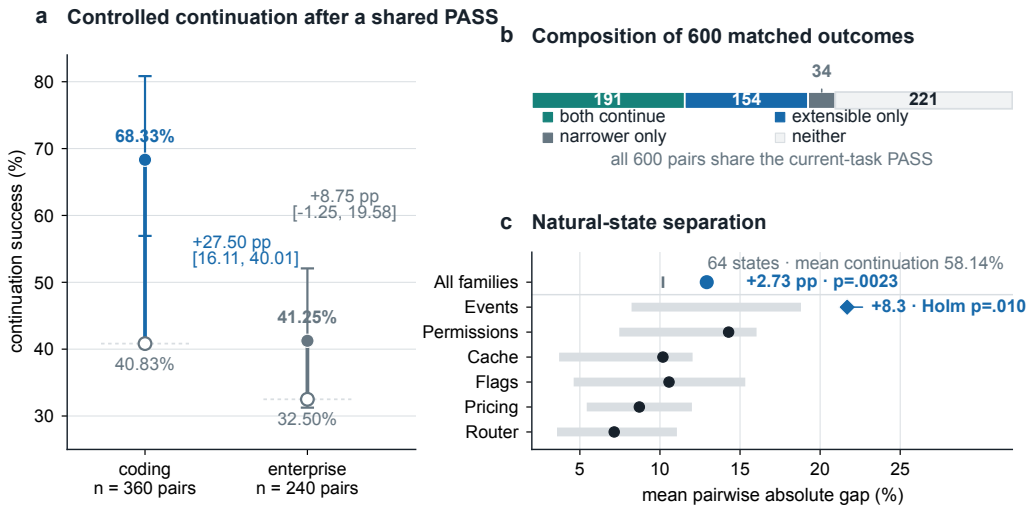

Figure 2: **A shared PASS hides structured continuation differences.** Panel a shows continuation success after the same current-task verdict for extensible (filled) and narrower (open) states; capped lines are hierarchical 95% confidence intervals for the extensible-minus-narrower difference, anchored at the observed narrower level. Coding separates, whereas the enterprise interval crosses its reference level. Panel b gives the exact composition of all 600 matched pairs by outcome cell. Panel c shows pooled and family-level natural-state separation against randomization nulls; thick grey bars are family-specific null 95% intervals.

draws nonpositive; all five executors and all six task families have positive point estimates; leave-one-model-out and leave-one-family-out gaps stay within 22.92–31.33 points. The complementary 240-pair enterprise arm shows the same direction: 41.25% versus 32.50%, an 8.75-point difference with hierarchical interval [-1.25, 19.58] and cluster-randomization $p = 0.052$.

Naturally produced states reproduce the effect without a hand-designed contrast. The five producers yielded 64 unique successful states with an equal-state mean continuation score of 58.14%. The primary L^1 endpoint is mean pairwise absolute separation, 12.92 points against 10.19 under the crossed null, leaving a corrected **2.73-point gap** ($p = 0.0023$; Figure 2c); the confirmatory $\mathcal{I}_2$-aligned noise-corrected excess variance is 0.00649 ($p = 0.00050$). Events provides the strongest family-specific separation, with a corrected 8.3-point gap (Holm-adjusted $p = 0.0102$).

Designed and model-produced states both witness continuation insufficiency. The next question is what the missing information costs a decision.

5.2 Decision: storing only the verdict has a measurable price

The frozen decision panel prices what a success-only record discards. Each repository contributes a historical maintainer state and one accepted alternative generated under an assigned design strategy without future information. Across 144 heldout executions in 18 repositories, a rule frozen from development continuations reaches 37.50% success against 24.31% under verdict-only uniform choice: a **13.19-point reduction in record-only regret** (repository-bootstrap 95% interval [5.56, 21.53]; exact random-policy tail $p = 0.0039$; Figure 3a). Eight repository effects favor retaining continuation information, ten tie, and none reverse. The rule attains the realized-panel oracle; the mechanism analysis below asks which state information produced that value.

History provides the strongest operational comparator and explains the primary contrast. Selection chooses the historical state in 16/18 repositories and an outcome-equivalent alternative in the other two, exactly matching the historical-state policy at 37.50%. For calibration, uniform tie-breaking achieves 24.31%, the candidate-only policy 11.11%, and shortest patch 34.72%; the selected–shortest contrast is 2.78 points ([0.00, 8.33]; exact two-sided $p = 1.0000$). Thus the experiment establishes

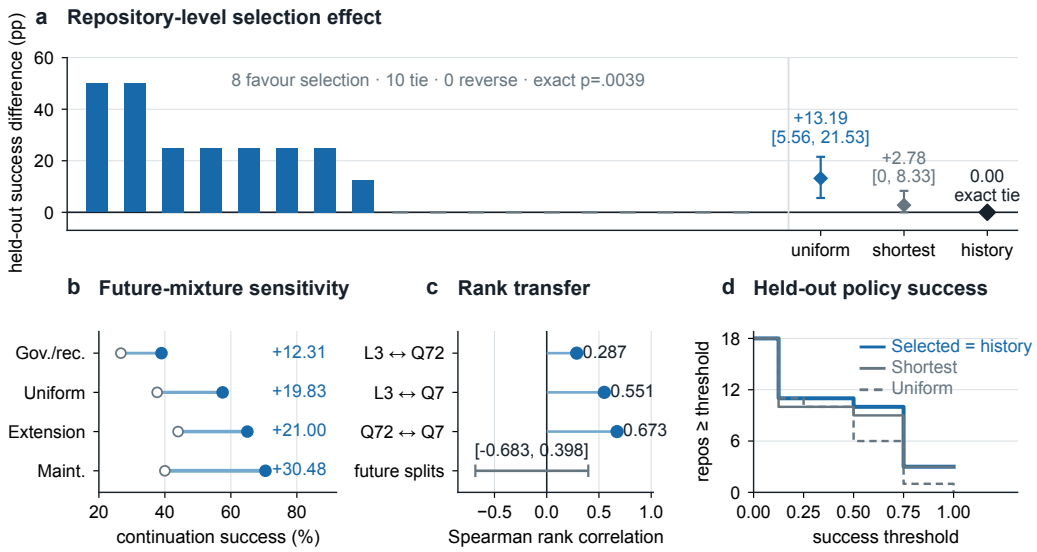

Figure 3: **Continuation information prices verdict-only choice and reveals context-indexed rankings.** Panel a gives selected-minus-uniform effects across 18 frozen repositories and repository-bootstrap contrasts with uniform, shortest patch, and history. Panel b shows four unordered future mixtures. Panel c gives consumer rank agreement (L3: Llama-3 8B; Q72/Q7: Qwen2.5 72B/7B) and the future-split range. Panel d gives policy profiles; selected and historical coincide.

the cost of deleting state identity, not an advantage over a deployment that already retains history. The mechanism is concrete: historical replay is conflict-free by construction, whereas 48/72 candidate executions stop at a source conflict before a future consumer runs (Figure 3d).

The preference itself is context-indexed. In the natural-state panel, consumer-specific rankings have pairwise Spearman correlations of 0.287, 0.551, and 0.673, and future-split state-rank correlations span $[-0.683, 0.398]$ (Figure 3c). In the controlled panel, reweighting outcomes across future types moves the extensible-minus-narrower gap from 12.31 points under governance and recovery futures to 30.48 points under maintenance futures, with intermediate mixtures at 19.83 and 21.00 (Figure 3b). The decision panel never reverses, but Jsonpickle does: history replay favors the historical state and a common snapshot favors the alternative (Section 5.3).

Takeaway. Compressing these states to PASS incurs 13.19 points of choice regret relative to retaining continuation evidence. The equality with the historical baseline identifies history compatibility—not a universal state ranking—as the dominant missing information on this panel.

5.3 Mechanism: replay controls separate transition compatibility from downstream utility

Three repositories make the failure stages auditable (Figure 4). In Kinto, the candidate state diverges at the transition itself: strict replay stops at a source conflict and the three futures are never reached, so nothing downstream is observed. In Moto, replay succeeds and the candidate then produces a benefit event on a future the historical state misses. Jsonpickle is the running example of Section 2: with history replay removed by a common snapshot, the candidate still differs downstream. Together the cases separate transition incompatibility, post-transition benefit, and common-snapshot downstream utility, three mechanisms that a scalar state score would conflate.

Recovery makes the first stage quantitative through partially identified preservation bounds. Raising the budget from $B = 0$ to $B = 3$ raises the certifiable-recovery lower bound by 23.91 points for Codex GPT-5.6 and 13.04 for Claude Opus 4.8; gains occur in two of eight repositories for Codex and one for Claude, and none declines (Figure 5a,c). At $B = 10$, empirical preservation is $[56.52, 100.00]\%$ for Codex and $[43.48, 47.83]\%$ for Claude. By Equation 5, interval width records

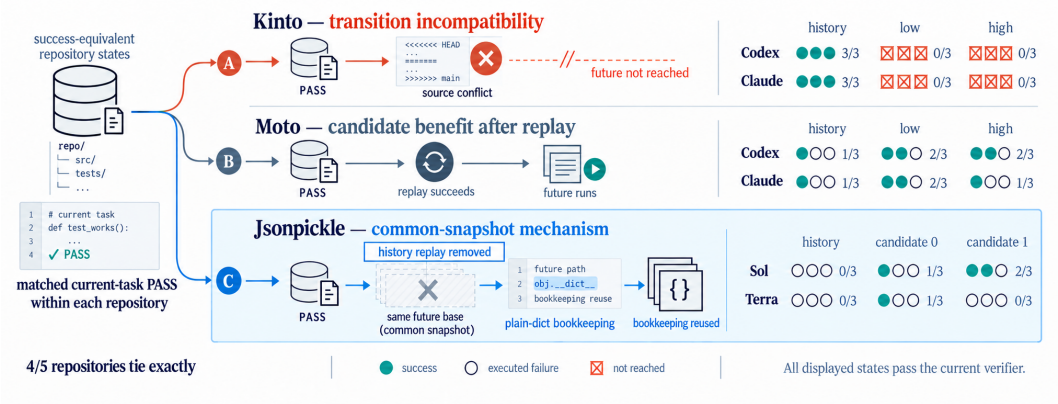

Figure 4: **One shared PASS, three places where continuations diverge.** Filled discs: successful downstream executions. Open rings: executed failures. Crossed squares with a 0/3 ledger: futures never reached because strict replay stopped at a source conflict, which are not executed failures. Rows are future consumers for Kinto and Moto and executors for Jsonpickle. Denominators and claim boundaries appear in Appendix Table 5.

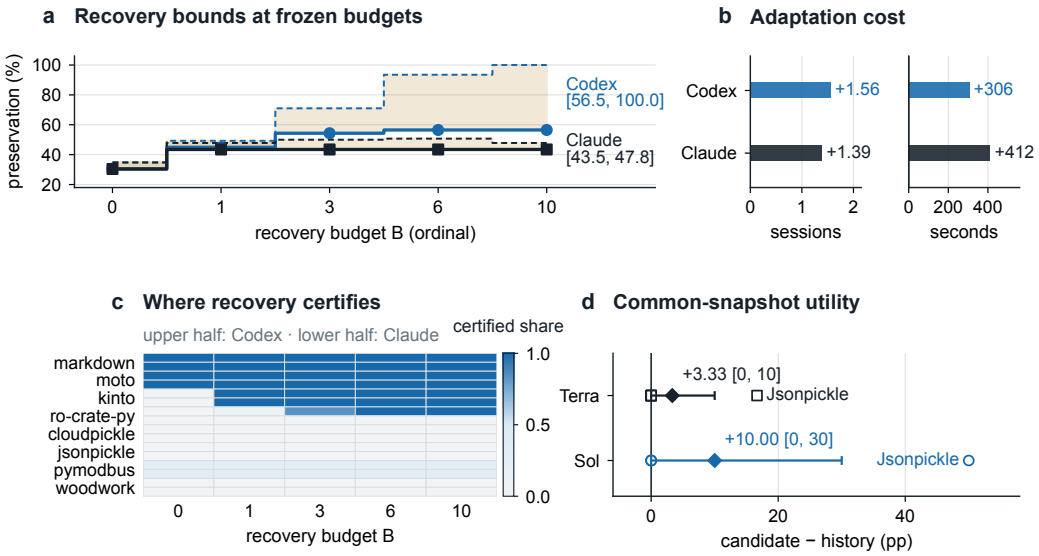

Figure 5: **Repository continuation separates certification, cost, and utility.** Panel a reports certifiable lower (solid) and procedural upper (dashed) preservation bounds; the amber verifier-missingness band is an identified set, not a confidence interval. Panel b gives candidate-minus-historical adaptation cost at matched replay. Panel c localizes certifiable recovery by repository, budget, and consumer (upper half: Codex; lower: Claude). In panel d, open markers are repository effects and diamonds are aggregate differences with intervals.

verifier availability, and every transported verifier that executes preserves the prior behavior. Budget curves, repository strata, and agent-mediated recovery are in Appendix E.

When both state types are driven to the same endpoint, the comparison identifies adaptation cost. In the matched $B = 10$ control both reach 100% replay, while candidates require 1.56 additional sessions and 306 seconds for Codex and 1.39 sessions and 412 seconds for Claude (Figure 5b). For both consumers, six repository time differences are positive and two tie, with none negative (two-sided exact sign-flip $p = 0.03125$); downstream-success intervals span zero.

Common-snapshot execution provides a replay-free mechanism witness by eliminating transition asymmetry. All 90 sessions preserve the prior verifier. Sol is the `gpt-5.6-sol` Codex executor used above; Terra is the otherwise matched `gpt-5.6-terra` executor. Their pair-weighted differences are +10.0 points [0, 30] and +3.33 points [0, 10], respectively (Figure 5d). The separation is localized to Jsonpickle, where successful future trajectories compose through the alternative's reference primitive.

Takeaway. Transition and history compatibility explain most of the decision effect: 48 of 72 candidate executions stop before any future task runs. The common-snapshot panel then verifies one downstream Jsonpickle mechanism after replay is removed.

Realized continuation remains the calibration target. Tested judges are position-sensitive (40–80% swap consistency); beyond producer context, state features add $+0.0056$ $[-0.0243, 0.0361]$; and prospective structure features improve average precision over patch size by $+0.003$ $[-0.200, 0.131]$ (Appendix F). Executed continuation therefore remains the reference against which cheaper proxies must be validated.

6 Related Work and Positioning

What agent evaluation measures. Current benchmarks execute immediate evaluation in interactive environments and mutable tools (Liu et al., 2024; Trivedi et al., 2024; Yao et al., 2025; Lu et al., 2025), and stateful enterprise, memory, and coding settings (Microsoft, 2026; Hu et al., 2026b; Shen et al., 2026; Wang et al., 2025); harness choices can move scores (Kapoor et al., 2026). They observe one episode or accumulating trajectory, not several states after the same accepted task. Unlike state abstraction and predictive state representations for a known decision problem (Li et al., 2006; Ferns et al., 2004; Littman et al., 2001), we audit record validity when futures and executors vary; Appendix H positions the full benchmark set.

Valuing futures. Safety work studies side effects (Amodei et al., 2016), option preservation (Krakovna et al., 2018; Turner et al., 2020; Krakovna et al., 2020), and proxy misspecification (Lindner et al., 2021). Foresight value compares locally successful states within one fixed manipulation chain (Liu et al., 2026); recoverability tests one delayed objective under an edit budget (Anonymous Authors, 2026). Our novelty is not the premise that future tasks have value. It is a counterfactual audit of whether an accepted *record preserves that value*: condition on one verdict, assign identical declared continuations to distinct states, and decompose any divergence into transition, preservation, utility, and cost (Appendix H).

7 Discussion and Conclusion

This work identifies a measurement failure in persistent-agent evaluation: a shared success verdict can collapse states with different continuation value. Continuation insufficiency formalizes the information lost by this compression, and matched-success evaluation makes it observable by assigning identical futures and separating transition, preservation, downstream utility, and cost. Across controlled pairs, naturally produced states, and frozen repository continuations, the omitted information is reproducible and large enough to change state selection. On the decision panel, retaining continuation evidence avoids 13.19 points of verdict-only regret and recovers the historical-state baseline. Replay and recovery locate that value in history compatibility; a common-snapshot mechanism witness exposes downstream divergence after replay is removed.

The practical implication is direct: continued-interaction benchmarks should retain multiple accepted states, declare continuation conditions, and report where future value is lost instead of recording success alone. The current decision panel uses alternatives from one producer, and its common-snapshot mechanism is concentrated in Jsonpickle; broader producer-diverse panels and validated low-cost predictors are the next extension. Success is therefore not the end of measurement—the environment it leaves behind is part of the result.

REFERENCES

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*, 2016. URL https://arxiv.org/abs/1606.06565.

Anonymous Authors. Evaluating LLM agents beyond local edit validity. Anonymous ACL submission on OpenReview, 2026. URL https://openreview.net/pdf?id=bC0mNJcci4. Under review; OpenReview forum bC0mNJcci4.

Parth Asawa, Christopher M. Glaze, Gabriel Orlanski, Ramya Ramakrishnan, Benji Xu, Asim Biswal, Vincent Sunn Chen, Frederic Sala, Matei Zaharia, and Joseph E. Gonzalez. Continual learning bench: Evaluating frontier AI systems in real-world stateful environments. *arXiv preprint arXiv:2606.05661*, 2026. URL https://arxiv.org/abs/2606.05661.

Jialong Chen, Xander Xu, Hu Wei, Chuan Chen, and Bing Zhao. SWE-CI: Evaluating agent capabilities in maintaining codebases via continuous integration. *arXiv preprint arXiv:2603.03823*, 2026. URL https://arxiv.org/abs/2603.03823.

Alexander D'Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D. Hoffman, et al. Underspecification presents challenges for credibility in modern machine learning. *Journal of Machine Learning Research*, 23(226):1–61, 2022.

Gangda Deng, Zhaoling Chen, Zhongming Yu, Haoyang Fan, Yuhong Liu, Yuxin Yang, Dhruv Parikh, Rajgopal Kannan, Le Cong, Mengdi Wang, Qian Zhang, Viktor Prasanna, Xiangru Tang, and Xingyao Wang. SWE-Milestone: Evaluating AI agents on continuous software evolution. *arXiv preprint arXiv:2603.13428*, 2026. URL https://arxiv.org/abs/2603.13428.

Norm Ferns, Prakash Panangaden, and Doina Precup. Metrics for finite markov decision processes. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, pp. 162–169, 2004. URL https://www.cs.mcgill.ca/~prakash/Pubs/mico.pdf.

Thibaud Gloaguen, Niels Mündler, Mark Müller, Veselin Raychev, and Martin Vechev. Coding agents don't know when to act. *arXiv preprint arXiv:2605.07769*, 2026. URL https://arxiv.org/abs/2605.07769.

Deyao Hong, Yizhe Chi, Wenyi Li, Xiaoqiu Wang, Mingju Gao, Kaisen Yang, Bingxiang He, Youjie Zheng, Calvin Xiao, and Qinhuai Na. SWE Refactor Bench: Can coding agents complete a long-horizon, whole-repository stack migration? *arXiv preprint arXiv:2608.23564*, 2026. URL https://arxiv.org/abs/2608.23564.

Qi Hu, Yifeng Tang, Qinghua Wang, Lanyang Zhao, Pengji Zhang, Yuhao Qing, Xin Yao, Dong Huang, Lin Zhang, and Zhuoran Ji. SABER: Benchmarking operational safety of LLM coding agents in stateful project workspaces. *arXiv preprint arXiv:2606.01317*, 2026a. URL https://arxiv.org/abs/2606.01317.

Yuanzhe Hu, Yu Wang, and Julian McAuley. Evaluating memory in LLM agents via incremental multi-turn interactions. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://arxiv.org/abs/2507.05257.

Sihang Jiang, Lipeng Ma, Zhonghua Hong, Keyi Wang, Zhiyu Lu, Tengfei Wang, Shisong Chen, Jinghao Zhang, Tianjun Pan, Weijia Li, Jiaqing Liang, and Yanghua Xiao. SEA-Eval: A benchmark for evaluating self-evolving agents beyond episodic assessment. *arXiv preprint arXiv:2604.08988*, 2026. URL https://arxiv.org/abs/2604.08988.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R. Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=VTF8yNQM66.

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 *The Fourteenth International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2510.11977.

Victoria Krakovna, Laurent Orseau, Ramana Kumar, Miljan Martic, and Shane Legg. Penalizing side effects using stepwise relative reachability. *arXiv preprint arXiv:1806.01186*, 2018. URL https://arxiv.org/abs/1806.01186.

Victoria Krakovna, Laurent Orseau, Richard Ngo, Miljan Martic, and Shane Legg. Avoiding side effects by considering future tasks. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/dc1913d422398c25c5f0b81cab94cc87-Abstract.html.

Tue Le, Minh V. T. Thai, Dung Nguyen Manh, Huy Phan Nhat, and Nghi D. Q. Bui. SWE-EVO: Benchmarking coding agents in long-horizon software evolution scenarios. *arXiv preprint arXiv:2512.18470*, 2025. URL https://arxiv.org/abs/2512.18470.

Lihong Li, Thomas J. Walsh, and Michael L. Littman. Towards a unified theory of state abstraction for MDPs. In *Proceedings of the Ninth International Symposium on Artificial Intelligence and Mathematics*, 2006. URL https://thomasjwalsh.net/pub/aima06Towards.pdf.

David Lindner, Kyle Matoba, and Alexander Meulemans. Challenges for using impact regularizers to avoid negative side effects. *arXiv preprint arXiv:2101.12509*, 2021. URL https://arxiv.org/abs/2101.12509.

Michael L. Littman, Richard S. Sutton, and Satinder Singh. Predictive representations of state. In *Advances in Neural Information Processing Systems*, volume 14, 2001.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. Agentbench: Evaluating LLMs as agents. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=zAdUB0aCTQ.

Yuhan Liu, Xinyu Zhang, Litao Liu, and Abdeslam Boularias. Foresight residual RL for long-horizon robot manipulation with vision-language-action models. *arXiv preprint arXiv:2607.16506*, 2026. URL https://arxiv.org/abs/2607.16506. Accepted at IROS 2026.

Jiarui Lu, Thomas Holleis, Yizhe Zhang, Bernhard Aumayer, Feng Nan, Haoping Bai, Shuang Ma, Shen Ma, Mengyu Li, Guoli Yin, Zirui Wang, and Ruoming Pang. Toolsandbox: A stateful, conversational, interactive evaluation benchmark for LLM tool use capabilities. In *Findings of the Association for Computational Linguistics: NAACL 2025*, 2025. doi: 10.18653/v1/2025.findings-naacl.65. URL https://aclanthology.org/2025.findings-naacl.65/.

Chang Ma, Junlei Zhang, Zhihao Zhu, Cheng Yang, Yujiu Yang, Yaohui Jin, Zhenzhong Lan, Lingpeng Kong, and Junxian He. Agentboard: An analytical evaluation board of multi-turn LLM agents. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://openreview.net/forum?id=4S8agvKjle.

Sina Mavali, David Pape, Jonathan Evertz, Samira Abedini, Devansh Srivastav, Thorsten Eisenhofer, Sahar Abdelnabi, and Lea Schönherr. No more, no less: Task alignment in terminal agents. *arXiv preprint arXiv:2605.12233*, 2026. URL https://arxiv.org/abs/2605.12233.

Microsoft. STATE-Bench: Benchmark for enterprise workflows. Official benchmark repository, 2026. URL https://github.com/microsoft/STATE-Bench. Accessed 2026-08-31.

Bardia Mohammadi, Lars Klein, Aman Chadha, Akhil Arora, and Laurent Bindschaedler. The working set of a coding agent: Coherence debt in repository-scale tasks. *arXiv preprint arXiv:2608.16630*, 2026. URL https://arxiv.org/abs/2608.16630.

Gabriel Orlanski, Devjeet Roy, Alexander Yun, Changho Shin, Alex Gu, Albert Ge, Dyah Adila, Nicholas Roberts, Frederic Sala, and Aws Albarghouthi. Slopcodebench: Benchmarking how coding agents degrade over long-horizon iterative tasks. *arXiv preprint arXiv:2603.24755*, 2026. URL https://arxiv.org/abs/2603.24755.

David Lorge Parnas. Software aging. In *Proceedings of the 16th International Conference on Software Engineering*, pp. 279–287, 1994. doi: 10.1109/ICSE.1994.296790. URL https://doi.org/10.1109/ICSE.1994.296790.

D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-François Crespo, and Dan Dennison. Hidden technical debt in machine learning systems. In *Advances in Neural Information Processing Systems*, volume 28, 2015. URL https://proceedings.neurips.cc/paper/2015/hash/86df7dcfd896fcaf2674f757a2463eba-Abstract.html.

Haiyang Shen, Xuanzhong Chen, Wendong Xu, Yun Ma, Liang Chen, and Kuan Li. EvoCode-Bench: Evaluating coding agents in multi-turn iterative interactions, 2026. URL https://arxiv.org/abs/2605.24110.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. AppWorld: A controllable world of apps and people for benchmarking interactive coding agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024. doi: 10.18653/v1/2024.acl-long.850. URL https://aclanthology.org/2024.acl-long.850/.

Alexander Matt Turner, Dylan Hadfield-Menell, and Prasad Tadepalli. Conservative agency via attainable utility preservation. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pp. 385–391, 2020. doi: 10.1145/3375627.3375851. URL https://doi.org/10.1145/3375627.3375851.

You Wang, Michael Pradel, and Zhongxin Liu. Are "solved issues" in SWE-bench really solved correctly? an empirical study. In *Proceedings of the IEEE/ACM International Conference on Software Engineering (ICSE)*, 2026. arXiv:2503.15223.

Zhengren Wang, Rui Ling, Chufan Wang, Yongan Yu, Sizhe Wang, Zhiyu Li, Feiyu Xiong, and Wentao Zhang. Maintaincoder: Maintainable code generation under dynamic requirements. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2503.24260.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/5d413e48f84dc61244b6be550f1cd8f5-Abstract-Datasets_and_Benchmarks_Track.html.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2406.12045.

Dewu Zheng, Yanlin Wang, Ensheng Shi, Ruikai Zhang, Yuchi Ma, Hongyu Zhang, and Zibin Zheng. HumanEvo: An evolution-aware benchmark for more realistic evaluation of repository-level code generation. In *Proceedings of the 47th IEEE/ACM International Conference on Software Engineering*, 2025. URL https://arxiv.org/abs/2406.06918.

Chenyu Zhou, Xinyun Lu, Jiangyue Zhao, Jianghao Lin, Dongdong Ge, and Yinyu Ye. OR-Space: A full-lifecycle workspace benchmark for industrial optimization agents. *arXiv preprint arXiv:2605.28158*, 2026. URL https://arxiv.org/abs/2605.28158.

ETHICS STATEMENT

The experiments use open-weight and proprietary API language models, public benchmark repositories, and executable software artifacts. Repository histories are analyzed as software artifacts; no human-subject or private user data are intentionally included. Any released model weights, benchmark instances, and repository snapshots remain subject to their upstream licenses. The paper reports negative results and selection limitations to reduce the risk that a diagnostic prototype is mistaken for a deployable safety guarantee. The study does not claim that the measured effect is prevalent in deployed systems or that its diagnostic proxies are suitable for autonomous enforcement.

REPRODUCIBILITY STATEMENT

Code and frozen evaluation artifacts are available in the anonymous code repository (ID: `matched-success-continuation-evaluation-B117`). The snapshot contains pair manifests, protocol documents, raw JSONL outcomes (including every prospective replay, calibration, heldout, and recovery rollout), integrity hashes, runner and statistical scripts, and corrected analysis files. Its release manifest identifies upstream artifacts that cannot be redistributed and provides commands for recreating the headline values, tables, and figures without rerunning model inference.

AI ASSISTANCE DISCLOSURE

Language models were used for experiment orchestration, code generation and debugging, literature discovery, and manuscript drafting and editing. All reported quantitative results were produced by executable scripts and checked against stored artifacts. The authors directed the research decisions, reviewed the cited sources and claims, and take responsibility for the accuracy and integrity of the submission.

APPENDIX

A DERIVATIONS

Table 1: **Core notation.** Continuation quantities are indexed by the declared state and context distributions.

Symbol	Meaning	Symbol	Meaning
s, z	terminal state and stored record	$\mathcal{E}_z$	states sharing record z
μ_z, ν	state and context distributions	c	$(\mathcal{D}_F, \pi_F, g_F, B)$
$v(s,c)$	expected future outcome	$\mathcal{I}_2$	excess record-based Bayes risk
$d(c)$	matched-state value difference	$\widehat{V}$	finite-panel continuation estimate
M_F, E_F	transition and verifier availability	L_s, U_s	preservation lower and upper bounds
Q_s, K_s	downstream utility and cost	ρ_B	recovery policy at budget B

Projection identity in Equation 1. For fixed (z, c), write $\bar{v}_z(c) = \mathbb{E}_{\mu_z}[v(S,c)]$. Squared-loss projection and the conditional-variance decomposition give

$$\mathrm{Var}(G_F \mid z, c) = \mathbb{E}[\mathrm{Var}(G_F \mid S, c) \mid z, c] + \mathrm{Var}(v(S,c) \mid z, c).$$

The record-based and state-based Bayes risks are the left side and first term on the right, respectively. Integrating their difference over $c \sim \nu$ proves Equation 1. At $\epsilon = 0$ this is predictive mean-sufficiency under squared loss, not statistical sufficiency for the full conditional law of G_F.

Two-state witness in Equation 2. For the uniform distribution on $\{s_a, s_b\}$, the variance of the two continuation values at context c is $d(c)^2/4$. Averaging over ν gives the equality, and Jensen's inequality gives $\mathbb{E}[d(C)^2] \geq (\mathbb{E}[d(C)])^2$. A matched mean contrast therefore witnesses positive aliasing risk, but a null mean does not establish sufficiency because context-specific effects may cancel. A signed reference contrast, such as residual-state debt, is decision-specific, whereas $\mathcal{I}_2$ is a property of the record under (μ_z, ν).

Orthogonal continuation decomposition and the transferable share. Under the product measure $\mu_z \times \nu$ the continuation surface has the exact centered decomposition

$$v(s,c) = m + a(s) + \beta(c) + \gamma(s,c), \qquad \mathcal{I}_2 = \underbrace{\mathbb{E}[a(S)^2]}_{\sigma_a^2} + \underbrace{\mathbb{E}[\gamma(S,C)^2]}_{\sigma_\gamma^2}, \tag{6}$$

with $a(s) = \mathbb{E}_C[v(s,C)] - m$, $\beta(c) = \mathbb{E}_S[v(S,c)] - m$, and γ centered over either argument; hence $\mathbb{E}_S a = \mathbb{E}_C \beta = 0$ and $\mathbb{E}_S \gamma(S,c) = \mathbb{E}_C \gamma(s,C) = 0$. The ratio $\tau = \sigma_a^2/\mathcal{I}_2$ would classify the transferable share of state-associated variation, with $\tau = 1$ meaning an additive state effect across the declared contexts. We do not estimate τ, because separating Bernoulli noise from interaction requires prespecified replication; the rank and mixture shifts in Section 5.2 diagnose interaction directly instead.

Context-free regret in Equation 3. A context-free selector that always chooses s_a loses $(-d(C))_+$ relative to the context-aware oracle; always choosing s_b loses $d_+(C)$. Randomizing gives a convex combination whose minimum is attained at an endpoint. If the selector is permutation-invariant because it observes only the shared record, its mixing probability is $1/2$ and its regret is $\frac{1}{2}\mathbb{E}|d(C)|$.

Worst-case envelope. For a context family $\mathcal{C}$, $d_{\mathcal{C}}(s_i, s_j) = \sup_{c \in \mathcal{C}} |v(s_i,c) - v(s_j,c)|$ is a pseudo-metric and $D_{\mathcal{C}}(z) = \sup_{s_i, s_j \in \mathcal{E}_z} d_{\mathcal{C}}(s_i, s_j)$ is its fiber diameter. These bound adversarial separation but are not identified by a finite sampled panel. Enlarging $\mathcal{C}$ cannot decrease either quantity, whereas $\mathcal{I}_2(z; \mu_z, \nu)$ changes with the declared distributions.

Budgeted recovery profiles. Let ρ_B be a recovery policy with budget B; let $M_F(s, \rho_B)$ indicate that the transition reaches the future base, E_F that the frozen current-task verifier can be transported

and executed there, Y_F that it passes, H_F the latent event that the earlier accepted behavior is preserved (including when $E_F = 0$), and $K_F(s, \rho_B)$ the incurred cost. Define

$$\mathcal{R}_s(B) = \big(P_s(B), L_s(B), U_s(B), Q_s(B), K_s(B)\big),$$
$$P_s(B) = \Pr[M_F = 1], \qquad L_s(B) = \Pr[M_F E_F Y_F = 1],$$
$$U_s(B) = P_s(B) - \Pr[M_F E_F (1 - Y_F) = 1], \qquad Q_s(B) = \mathbb{E}[M_F H_F G_F(\Phi(\tilde{s}_{F,B}, F, \xi_F))]\,,$$

with $K_s(B) = \mathbb{E}[K_F]$ including failed attempts. A valid verifier pass supplies the lower bound when it implies $H_F = 1$; a valid verifier failure tightens the upper bound when it implies $H_F = 0$. Under these directional assumptions and the requirement that recovery reach the future base, $L_s(B) \leq \Pr[M_F H_F = 1] \leq U_s(B)$, which is Equation 5. Here L_s is certifiable recovery, whereas U_s counts verifier-unavailable completed replays as preserved unless an executed verifier fails. No executed prior-task verifier fails in the final sweep, so $U_s = P_s$ and $U_s - L_s$ consists entirely of unidentified outcomes caused by verifier unavailability. The recovery-only sweep estimates (P, L, U, K); tiers that proceed to a future task estimate Q. With $V_\lambda = Q - \lambda K$ and the interval for ΔQ spanning zero, even cost-free benefit is unidentified.

B Protocol and reproducibility

Prospective state bank. A preregistered repository-disjoint extension attempted to replace patch compatibility with actual future-agent rollouts. After validating 19/19 frozen current-task verifiers, one fixed Qwen3.6-35B-A3B producer generated 12 trajectories for each of 19 anchors. All 228 runs were operationally valid; 52 passed syntax, test-contamination, hidden current-task, regression, and metamorphic-verifier gates. Deduplication left 41 current-success hashes. Requiring at least three alternatives per anchor retained 35 states over Moto, Jsonpickle, Kinto, Pymodbus, and RO-Crate. Each anchor had one audit future and three heldout futures frozen before downstream evaluation.

Consumer calibration and isolation. The future consumer had to solve at least 30% of historical-gold controls, change code in at least 80% of rollouts, and produce fewer than 5% invalid actions. Codex GPT-5.6 solves 6/15 controls and Claude Opus 4.8 solves 9/15; both change code in 15/15 trajectories and record zero invalid actions. Containers are network-isolated, agents receive only the issue and repository interface, and hidden tests are installed after the agent exits. The excluded calibration consumer and its prospective gate are documented in Appendix G.

Sequential replay. We create a detached worktree at the earlier base, apply and commit the candidate state, and cherry-pick every first-parent commit through the frozen future base except the historical current-task integration. Merge commits are replayed relative to parent one and empty commits are skipped. Metadata-only conflicts accept the upstream version and are counted; source conflicts stop the strict continuation pipeline. On the audit future, 22/35 candidates replay and 13/35 encounter source conflicts: five Jsonpickle and eight Kinto. Three fresh repetitions of one high-severity state per repository reproduce the same conflicting commit and file in all 6/6 runs.

Frozen candidate ranking and heldout execution. Candidate selection is frozen before heldout execution. Within each anchor, candidates are ranked lexicographically by audit source-conflict indicator, metadata-conflict count, audit continuation failure, and patch churn; the minimum and maximum define low and high. For each of three heldout futures we run the historical, low, and high states. Codex obtains 7/15, 2/15, and 2/15 strict successes; Claude obtains 6/15, 2/15, and 1/15. The consumers disagree on only two of 45 arms. Twenty of 30 candidate arms per consumer stop at a source conflict; the remainder proceed to autonomous future-task execution. These selected rates are not prevalence estimates.

Common-snapshot panel. The five common-snapshot repositories are nested within the eight-repository recovery panel; the 18-repository decision panel is analyzed separately. Futures are selected using historical feasibility only, retaining candidate incompatibility as an outcome. Two frozen executors contribute 45 sessions each. Across 90 sessions, four repositories are uniformly solved and only Jsonpickle is state-dependent; candidate-level performance differs between consumers. The prospective amendment and stopped repository-disjoint extension appear in Appendix G.

Claim-to-artifact map. Table 2 names the frozen file that produces each headline value, so a reader can re-derive the number without rerunning model inference.

Table 2: **Claim-to-artifact map.** Paths are relative to the root of the anonymous code repository cited in the Reproducibility Statement. The release script `scripts/reproduce_headline_results.py` independently re-checks the controlled, sequential-decision, cross-family selection, recovery, and common-snapshot values from these files.

REPORTED VALUE	FROZEN ARTIFACT
Controlled coding 68.33% vs. 40.83% (**27.50 pp**)	`empirical_results/coding_scale_summary.json`
Controlled enterprise 41.25% vs. 32.50% (**8.75 pp**)	`empirical_results/enterprise_scale_summary.json`
Natural-state corrected pairwise gap **2.73 pp** ($p = 0.0023$)	`empirical_results/natural_states_reference_free.json`; `analysis/analyze_natural_states_reference_free.py`
Cross-family selection **+9.97 pp**, frozen gate not met	`prospective_rsd_v3/continuation_selection_analysis/analysis_yury_official.json`
Sequential decision **+13.19 pp**; selected = history	`prospective_rsd_v3/sequential_decision_heldout/analysis/sequential_selection_results.json`; `prospective_rsd_v3/sequential_decision_heldout/scripts/analyze_sequential_decision_heldout_v1c.py`
Sequential replay **22/35** automatic, **13/35** source conflicts	`prospective_rsd_v3/raw_outcomes/sequential_replay_metadata_sensitivity.jsonl`; `prospective_rsd_v3/continuation_freeze/sequential_replay_v4.json`
Certifiable recovery at $B = 10$: **56.52%** Codex, **43.48%** Claude	`prospective_rsd_v3/recovery_curve_analysis/recovery_curve_cluster_results.json`
Matched $B = 10$ control, **856** consumer-matched rows	`prospective_rsd_v3/control_continuation_results/paired_formal_856.jsonl`
Common snapshot **+10.0 pp** (Sol), **+3.33 pp** (Terra)	`prospective_rsd_v3/symmetric_sol_repeats/analysis_45.json`; `prospective_rsd_v3/symmetric_terra/analysis_45_terra.json`
Jsonpickle downstream divergence at common snapshot	`prospective_rsd_v3/jsonpickle_mechanism/jsonpickle_mechanism_analysis.json`

C CONTROLLED AND NATURAL-STATE EXPERIMENTS

Table 3 states the design logic of the five-tier evidence ladder introduced in Section 4, and Table 4 preserves the denominators and claim boundaries behind Section 5.1.

Model identities. Controlled executors and natural-state producers use Gemma-2 9B, Llama-3 8B, Phi-3.5, Qwen2.5 7B, and Qwen2.5 72B. Natural-state continuations use the Llama-3 8B, Qwen2.5 7B, and Qwen2.5 72B consumers (L3, Q7, and Q72 in Figure 3).

Table 3: **Each tier removes one alternative explanation.** All tiers condition on current-task success; later tiers trade experimental control for repository realism.

TIER	WHAT IT CONTROLS FOR	QUESTION IT ANSWERS
Controlled pairs	Current success and shared hidden futures	Can one record hide continuation differences? (RQ1)
Natural states	Hand-designed state contrast	Do model-produced states differ too? (RQ1)
Frozen selection	Post-outcome selection of a policy	What does a verdict-only choice cost? (RQ2)
Replay and recovery	Silent transition failure	How much of that cost is transition compatibility? (RQ3)
Common snapshot	History and replay asymmetry	Does a downstream effect remain without replay? (RQ3)

The controlled tier pairs an extensible and a narrower accepted state per task across six coding families and SQLite enterprise workspaces, with five executors supplying 600 paired comparisons. The natural tier draws on five producers whose 78/90 passing attempts deduplicate to 64 unique successful states, each attempted by three independent consumers on four hidden futures. All current-success gates are executed, natural states are deduplicated, and hidden futures run from fresh clones.

Table 4: **Matched-success evidence across controlled and natural settings.** Percentages are continuation success unless noted; repository examples establish existence rather than prevalence.

SETTING	UNIT	OUTCOME	INTERPRETATION
Controlled coding	360 pairs	68.33% extensible vs. 40.83% narrow	**27.50 pp**; 95% CI [16.11, 40.01]
Controlled enterprise	240 pairs	41.25% vs. 32.50%	**8.75 pp**; CI [-1.25, 19.58]
Natural coding states	64 states	Equal-state mean 58.14%; corrected pairwise gap 2.73 pp	$p = 0.0023$; one of six families survives Holm correction
Task-instance-disjoint confirmation	9 rule-positive cases	**3/9** direct continuation wins	Frozen low-coverage selector; repositories may recur
Same-producer repository	9 states	**0/5–5/5** future patches mergeable	Known-action compatibility witness
Prospective sequential audit	35 states	**22/35** automatic replay; 13/35 source conflicts	State construction checked under real history

Discovery cases establish mechanisms, not prevalence. The discovery procedure searches temporal issue pairs for later maintainer patches that remove or replace source added by an earlier patch. The actual earlier maintainer solution is the historical state; a future-derived alternative prepays the later structural change while remaining valid for the earlier task. After correcting the anchor-feasibility gate, one of nine screened cases (PyDESeq2) is excluded because no state satisfies both task oracles; all eight remaining cases are valid direct counterfactual-only wins, with exploratory one-sided sign value $p = 0.0039$. This rate is intentionally not a prevalence estimate: future overlap selects the candidates and the counterfactual uses future information. The audit also shows why executable gates matter. In PyBaMM, blindly applying future code breaks the earlier task and a compatibility alias is required; raw future-patch applicability is invalid because correct prepayment can make that patch conflict. Across selected prepayment cases, remaining maintainer patch size falls from 956 to 177 lines (81.5%) and source churn from 418 to 54 (87.1%). These are descriptive totals for selected cases.

Task-instance-disjoint confirmation. The frozen confirmation excludes all 400 discovery task instances and applies the unchanged repayment rule before executing outcomes. Of 100 selected pairs, 73 pass historical validity and nine trigger the rule; all nine are executed without manual outcome filtering. Three are strict direct continuation wins (3/9), with a Wilson 95% interval of 12.1–64.6%. Among five candidates satisfying every anchor-feasibility and current-task gate, three are direct wins and two are compatible partial repayments. This confirms mechanism existence outside the discovery task instances, but repositories can recur and the rule has low coverage. A separate audit of 200 heldout repositories finds no cases under the exact same-file rule, which is a selector-coverage failure rather than evidence that those repositories contain no continuation effect.

Multi-future and same-producer construction. The multi-future study searches for anchors followed by three to five temporally ordered tasks connected through implementation files. It finds 437 candidate sequences and freezes one per repository for 50 repositories, comprising 233 anchor-to-future pairs; 31 sequences pass every historical-validity gate and 40 contain at least three valid future tasks. This establishes coverage, not an effect. A stricter executable follow-up begins from confirmed anchors and requires clean deltas to be constructible at the anchor and propagatable to every future base; three anchors and eight futures survive. Three futures are counterfactual-success/historical-failure cases, none favor the historical state, and five fail from both states (one-sided exact $p = 0.125$). Merge-aware recovery cost has the same pattern: three positive reductions, five exact ties, no counterfactual-worse case. Aggregate remaining patch size falls from 365 to 277 lines (24.1%)

and source churn from 160 to 119 (25.6%). The five ties prevent a broad maintainability claim: an effect materializes when a future traverses the affected abstraction, interface, or dependency, and nearby tasks can be unaffected. The same-producer experiment asks whether one model can generate accepted states with different futures. Early versions failed frozen feasibility gates; V3.2 explicitly marks the reference diff as present and counts a candidate only after the Git tree changes. Two independent Qwen2.5-72B seed blocks attempt 16 trajectories for a dataclasses-jsonschema anchor, nine pass the current task and produce nine unique post-reference deltas, and those plus two Werkzeug candidates are propagated to real successors.

Recoverability and accumulation are separate quantities. A continuation gap does not determine how easily state equivalence can be restored, so we report repair probability, post-repair continuation value, and repair cost separately. Qwen2.5-7B produces 5 verified repairs in 12 attempts and Qwen2.5-72B produces 6 in 12; given a verified repair, continuation success is 95.00% and 91.67%, while across all attempts post-repair continuation is 64.58% and 75.00%. Mean model-edit costs are 0.711 and 0.629 against an exact-rollback cost of 0.545. From mutation dose 20 to 100, continuation success falls by 50.0 points for Llama-3-8B and 37.5 points for both Gemma-2-9B and Phi-3.5-mini, while size-matched controls remain near ceiling and both Qwen models remain at 100%. The evidence supports model- and task-contingent accumulation, not a monotonic universal law.

D SELECTION AND CONTEXT SENSITIVITY

Cross-family selector: a failed prespecified gate. The selector uses only the earlier four-future by three-consumer development panel. Within each family it chooses the current-success state with the highest equal-cell continuation mean, with SHA256 breaking ties. Before opening outcomes we froze two previously unexecuted futures per family, three repeats, a new Qwen3-Coder-30B-A3B consumer, the exact random-choice test, and the requirement that all 384 keys be present exactly once; the static comparator chooses the shortest source. Every submitted continuation preserves the current verifier. Selection yields 91.67% versus 81.70% for random choice (+9.97 points, interval [-11.10, 35.28], exact $p = 0.2381$) and 69.44% for the shortest state. The selector attains the post-hoc heldout maximum in five of six families, but random choice often reaches that maximum too, because Cache and Router are universal ties and 13/15 Permissions states are perfect; its five-hit oracle count is therefore unremarkable under random choice ($p = 0.370$). As an explicitly exploratory analysis after the primary gate was opened, development continuation value correlates with heldout value within family ($r = 0.378$, within-family permutation $p = 0.00029$); controlling family and producer strengthens this to $r = 0.532$ ($p = 0.00867$), with leave-one-family-out correlations of 0.372–0.647. The prespecified primary endpoint nevertheless remains unsupported, and this analysis does not establish a selection policy.

Frozen repository decision panel. The independent sequential panel was frozen before any development or heldout future ran. Alternatives are produced from the committed reference solution under one of four frozen design strategies (invariant refactor, special-case removal, helper extraction, data-driven normalization) with no access to Git history or future information; each must change implementation files and pass the frozen current-task tests. After two candidate-only feasibility rounds, 16 seed repositories had an eligible alternative (Codex GPT-5.6 in 12, Qwen3-Coder-30B-A3B in four); a frozen reserve pool of three repositories added PyPika and Pyanalyze (Pyccel yielded no successful candidate), reaching the prespecified minimum of 18. The repositories are Amaranth, Covalent, Curator, cxxheaderparser, dataclasses-jsonschema, mkdocs-include-markdown-plugin, OpenMDAO, PyBaMM, Pyanalyze, pyfakefs, PyPika, pyserde, PySTAC, python-semver, Rich, Robocop, Traits, and unyt. The rule ranks the two states by mean development joint success, then fewer mean tool calls, smaller source diff, and SHA256. The panel contains one historical and one current-success agent state for each repository, with development and heldout futures disjoint within repository. The policy uses two development executions per state; evaluation uses two heldout futures and two fresh executions per state–future pair. All 144 keys are present exactly once, all 72 historical hidden-test patches install, and there are no setup or operational failures. Candidate history replay succeeds in 24/72 executions and hidden future tests install in 22/72; the remaining 48 source conflicts are prespecified strict-replay continuation failures, that is, recovery requirements rather than evidence of irreversibility. The primary selected-minus-uniform contrast is +13.19 points (95% interval [5.56, 21.53], exact one-sided $p = 0.0039$). The selector and historical default both achieve

Table 5: **Frozen decision, recovery, and common-snapshot panels.** Intervals resample the stated model or repository units. Recovery lower–upper pairs are empirical identified sets caused by verifier availability, not confidence intervals.

PANEL	UNIT	PRIMARY RESULT	BOUNDARY / CONTROL
Cross-family selection	384 runs, 6 families	91.67% selected vs. 81.70% random; **+9.97 pp** [-11.10, 35.28]	Exact $p = 0.2381$; frozen gate not met
Sequential decision	144 runs, 18 repos	37.50% selected vs. 24.31% uniform; **+13.19 pp** [5.56, 21.53]	Exact $p = 0.0039$; equals historical default (37.50%)
Recovery, Codex	46 arms, 8 repos	Certifiable lower endpoint 30.43% ($B = 0$) to **56.52%** ($B = 10$)	At $B = 10$, preservation is [56.52%, 100.00%]
Recovery, Claude	46 arms, 8 repos	Certifiable lower endpoint 30.43% ($B = 0$) to **43.48%** ($B = 10$)	At $B = 10$, preservation is [43.48%, 47.83%]
Matched $B = 10$ control	428 shared keys	Candidate cost: **+1.56 sessions** / +306 s (Codex); +1.39 / +412 s (Claude)	Both arms reach 100% replay; downstream-success intervals cross zero
Common snapshot	90 sessions, 5 repos	Sol: **+10.0 pp** [0.0, 30.0]; Terra: +3.33 pp [0.0, 10.0]	Prior task preserved in 90/90; effect concentrated in Jsonpickle

37.50%; shortest patch achieves 34.72%. Conditional on successful candidate replay (24 executions in nine repositories), the candidate succeeds in 8 and the historical state in 13 of the same 24 keys: three repositories favor history, six tie, and none favor the candidate, so the historical advantage is not only a conflict count. Only PyBaMM distinguishes selection from shortest patch, so that secondary contrast has no exact-test support. The pattern is consistent with development continuation detecting transition incompatibility; it does not isolate a general state-quality predictor from historical-state identity.

E REPLAY, RECOVERY, AND COMMON SNAPSHOT

Table 5 consolidates the frozen decision, recovery, and common-snapshot endpoints, including the identified sets reported in Section 5.3.

Repository strata and thresholds. Recovery thresholds differ by repository, and realized cost is nonmonotone because B caps sessions rather than normalized compute. At $B = 10$ Codex certifies all three repeats for 26/46 arms (78/138 executions) and completes another 60 executions whose frozen test patch is inapplicable, with no transported verifier running and failing; Claude certifies 20/46 arms and completes two more with unavailable verifiers. The $B = 0$ to $B = 3$ gains come from Kinto and, for Codex, RO-Crate. Procedural completion, certified preservation, and future utility remain distinct endpoints. Per-anchor outcomes expose both signal and limits: Kinto is 3/3 from historical and 0/3 from either alternative for both consumers; Jsonpickle is 1/3 versus 0/3 for both; Moto is 1/3 historical versus 2/3 low for both, a benefit event, with high at 2/3 for Codex and 1/3 for Claude; Pymodbus is 0/3 for every state; historical RO-Crate is 2/3 for Codex and 1/3 for Claude versus 0/3 alternatives. The low/high ranking therefore fails as a general critic even though the historical–alternative contrast replicates.

Agent-mediated recovery. For each of the 20 alternative arms that stop at a source conflict, the harness leaves the cherry-pick in progress and starts a fresh session of the same consumer with the conflicting commit and unmerged paths. The recovery agent may inspect history and run tests but may not reset, restore, rebase, abort, or skip. The harness verifies the resolution, resumes replay, runs the current-task diagnostic when applicable, and then executes the byte-identical future protocol. Under the exploratory ten-round pilot, Jsonpickle alternatives encounter six to eight successive conflicting commits; Codex recovers 7/7 arms (63.6 tool calls and 15.6 minutes per arm) with 4 later successes, while Claude recovers 6/6 (128.2 tool calls and 32.7 minutes) with 3 later successes. The

2 regression failures per consumer occur on one future that the historical state also fails, and one recovered alternative solves a future missed by the historical state. The current-task test patch is inapplicable at the future base for Jsonpickle and Pymodbus, so the end-of-pipeline prior-task check and strict success definition remain the final gates. A source conflict denotes an automatic-replay recovery requirement, not irreversibility or failed agent-mediated repair.

Final independent-budget curve. We froze budgets $\{0, 1, 3, 6, 10\}$, three independent repeats for every positive budget, two alternatives and three futures per repository, and a recovery-only endpoint. The panel replicates the five prospective-bank repositories and adds Cloudpickle, Python-Markdown, and Woodwork, for which no recovery outcome had previously been opened. Per consumer, the final analysis contains 598 rows and 46 complete arms; every arm has one $B = 0$ run and three fresh runs at each positive budget. It reports replay failure, replay completion with verifier unavailability, and replay completion with verifier pass or failure separately. Appendix G records the pilot-only rerun and the pre-analysis exclusions that reduce 624 unique rows to 598 per consumer.

Discrete compatibility loss. Repository experiments also expose discrete compatibility loss. If a future maintainer patch applies cleanly to one state but not another, the states differ in compatibility with that known future action. Patch mergeability is not a complete measure of future-agent success or of the full action space, but it is an executable witness that current-success-equivalent states can constrain later maintenance differently.

F PROXY AND CRITIC AUDITS

Direct judges and future-blind probes. Order-balanced language-model judges cannot replace execution. Across the five tested judges, overall accuracy ranges from **70% to 90%** and model–domain cells range from 58.3% to 100%. Every judge favors the second option; accuracy is much higher when the narrower state appears first than when the reference appears first, and semantic consistency after swapping order ranges from 40% to 80%. A judge may still be useful as a component of an evaluator, but an unbalanced pairwise preference is not a trustworthy ground truth. The same-producer probe result is more direct: all 23 corrected V3.2 probe-score rows pass and receive the same score, so their correlations with future outcomes are undefined for lack of variance, yet the candidates differ sharply in later test preservation and patch mergeability. Ordinary future-blind verification certifies current validity without revealing continuation compatibility.

Apparent state-critic gains can be producer confounding. A task-conditioned kernel initially appears promising. On 768 state-task rows and 72 task-consumer selection groups it selects states with mean continuation success 0.632 versus a random expectation of 0.587, a paired gain of +0.0454 with 95% cluster-bootstrap interval [0.0215, 0.0698], positive across 120 regularization and split configurations and exceeding 498 of 500 state-feature permutation null draws. That result does not identify state-specific continuation value. A producer-context baseline alone gains +0.0274 over random; adding state features beyond producer context yields only +0.0056, interval [-0.0243, 0.0361]; and when candidate sets hold producer identity fixed the conditional critic scores -0.0028 versus random, interval [-0.0125, 0.0069]. Even pointwise AUC misleads: task-only prediction reaches 0.891 because some futures are easier, while adding state context changes it only to 0.894. A valid critic evaluation must hold the future task fixed, use within-choice-set selection as the primary metric, and control producer identity.

Prospective prediction is mostly patch exposure. The temporal corpus supports a modest prospective signal. On 148 discovery pairs with 19 natural-repayment labels, patch size and shape achieve average precision 0.293 and ROC-AUC 0.693 against prevalence 0.128; on the task-instance-disjoint split the signal replicates at AP 0.339 and AUC 0.673 against prevalence 0.123. Earlier issue text and added-code text do not improve it. Richer representations also fail: on the heldout split, repository-structure features achieve AP 0.241 and AUC 0.496, and their combination with patch size reaches AP 0.258 versus 0.255 for size alone, a difference of +0.003 with bootstrap 95% interval [-0.200, 0.131] and one-sided $p = 0.653$. Qwen2.5-14B frozen representations obtain AP 0.130 and AUC 0.480; adding size gives AP 0.131 and AUC 0.485. These labels are positive-unlabeled and measure later code repayment rather than direct continuation failure.

Why the tested shortcuts fail. The negative findings share one structure. Current tests observe the wrong time boundary; static features suppress the future task and executor; direct judges are order-sensitive; probes verify local validity but tie across divergent futures; task-conditioned models can learn task difficulty and producer identity; and patch exposure predicts selected repayment labels but not a general continuation effect. Continuation rollout is expensive, but it currently supplies the reference measurement against which cheaper proxies must be validated. This parallels warnings from impact regularization, where a convenient state-deviation measure should not be assumed to preserve the intended future value (Lindner et al., 2021). The proper learning target is conditional and counterfactual, with multiple states from the same producer evaluated on the same future tasks.

G Protocol amendments and execution history

This section records prospective changes, excluded pilots, operational interruptions, and stopped expansions. None was used to choose a reported scientific outcome.

Consumer sensitivity gate. A same-snapshot Qwen3.6 consumer solved 3/15 historical-gold controls and changed code in 5/15, failing the frozen gate before any heldout outcome was opened. We then froze Codex GPT-5.6 and Claude Opus 4.8 under the same criteria; their qualifying control results appear in Appendix B.

Common-snapshot amendment. The initial deterministic feasibility rule required the future maintainer patch and verifier to succeed from all three states. After 18/72 deterministic checks, and before any agent outcome, we recognized that this rule conditions on a post-treatment variable. The amended protocol selects only on historical feasibility and retains candidate incompatibility as an outcome. Protocols 25–28, timestamps, manifests, and all diagnostic rows are included in the supplement. Protocol 27 adds two Sol repeats per state; Protocol 28 freezes the otherwise identical 45-session Terra panel.

Stopped repository-disjoint expansion. A strict historical-only scan evaluated 137/137 future pairs from 29 repositories and retained 15 pairs in nine repositories. Together with the original five, this yielded 14 repositories and missed the 15-repository gate. A frozen broader scan then evaluated 31/31 structurally valid pairs from nine additional repositories and retained four pairs in three repositories, producing 17 historical-feasible anchors before candidate generation. We froze two current-task-only Sol attempts for each of 12 new anchors and required successful candidates in at least 10. EvalML, Curator, and Pyflakes each exhausted both attempts without a current-success candidate. Even perfect pending outcomes could therefore yield at most nine new anchors, or 14 total. We stopped once this upper bound made the frozen gate unattainable and did not open any downstream future outcome. The audit measures a construction bottleneck; it does not enlarge the five-repository effect sample.

Recovery pilot and harness correction. Amendment 15 froze the three-round recovery protocol before any recovery rollout. One harness pilot exposed a false negative in a repository-wide conflict-marker scan that matched reStructuredText underlines. We restricted the scan to unmerged files and excluded the pilot from all counts. Amendment 16 re-executed only arms that exhausted three rounds under a ten-round budget and is retained as pilot evidence, not as the final budget curve. Two Claude arms initially ended at an account usage limit; we quarantined and re-executed them under the identical protocol, and only the replacement rows are counted.

Final-curve exclusions and incomplete endpoints. The two final JSONL shards for each consumer contain 624 unique rows in total. Before analysis we excluded the same two Woodwork arms across all 13 budget–repeat keys (26 rows) because the image lacks the frozen Git reference, leaving 598 rows and 46 complete arms per consumer. The matched $B = 10$ control uses 428 keys observed for both consumers and does not impute Claude's nine missing endpoints. Per-arm outcomes, prompts, stream logs, and exclusion records are included in the supplementary package.

H EXTENDED POSITIONING AND BENCHMARK IMPLICATIONS

Episodic, stateful, and sequential agent evaluation. AgentBench evaluates agents across eight interactive environments, while AgentBoard adds within-trajectory progress diagnostics (Liu et al., 2024; Ma et al., 2024). AppWorld, τ-bench, ToolSandbox, and OSWorld instantiate mutable applications, databases, tools, or computer interfaces and score agents against environment state rather than text alone (Trivedi et al., 2024; Yao et al., 2025; Lu et al., 2025; Xie et al., 2024). The Holistic Agent Leaderboard shows why controlled harnesses matter: scaffolds, reasoning configurations, leakage, and evaluation bugs can materially alter task-level scores (Kapoor et al., 2026). These benchmarks strengthen the validity of immediate evaluation, but their primary unit remains an episode. STATE-Bench evaluates repeated reliability and agent learning in enterprise workflows (Microsoft, 2026); Continual Learning Bench and SEA-Eval evaluate performance across task streams, while MemoryAgentBench tests retrieval, test-time learning, long-range understanding, and selective forgetting (Asawa et al., 2026; Jiang et al., 2026; Hu et al., 2026b); HumanEvo and EvoCode-Bench evaluate evolving or multi-round coding tasks, with EvoCode-Bench providing 26 executable tasks over 227 rounds (Zheng et al., 2025; Shen et al., 2026). Our unit instead forks several states after the same accepted task and executes a common future set, isolating the contribution of the inherited external state. SABER is a close safety neighbor because it evaluates executable final project states after coding-agent actions, detecting current-episode violations from executed or recommended unsafe actions and from harmful workspace changes (Hu et al., 2026a); continuation insufficiency instead concerns a latent consequence, where the final state passes the present verifier but changes the success or adaptation cost of later tasks. Current-state safety checks and continuation evaluation measure different time boundaries.

Debt as a conditional contrast. The term *debt* follows a software-engineering lineage: software aging describes loss of value through failure to adapt and accumulated change (Parnas, 1994), and hidden technical debt in machine-learning systems identifies liabilities invisible to immediate performance metrics (Sculley et al., 2015). In our framework debt is not the primitive object. It is the positive part of a reference-relative continuation contrast under specified deployment conditions: negative values represent continuation benefit, and changing the reference, future distribution, executor, or cost can change the sign. This prevents static style or reference proximity from being promoted to ground truth.

Implications for persistent-agent evaluation. Immediate success remains necessary; it is simply incomplete when an agent leaves state for later use. A persistent-agent scorecard should report current success together with a continuation distribution over stated future tasks and executors. Where rollback or repair is possible, it should also report procedural completion, verifier availability, certifiable recovery, unidentified outcomes, and recovery cost. Collapsing these into one scalar is appropriate only when a deployment supplies defensible utility weights. The proposed unit complements existing stateful and lifecycle benchmarks: AppWorld, ToolSandbox, STATE-Bench, and SABER provide executable environments with mutable artifacts or databases, while OR-Space organizes a persistent multi-artifact lifecycle (Trivedi et al., 2024; Lu et al., 2025; Microsoft, 2026; Hu et al., 2026a; Zhou et al., 2026); FixedBench exposes unnecessary action and TAB exposes selective failures to follow necessary environmental cues (Gloaguen et al., 2026; Mavali et al., 2026). The software results refine the relation to maintainability: SWE Refactor Bench evaluates prescribed repository migrations and Coherence Debt studies stale or missing coupled facts in a coding agent's working set (Hong et al., 2026; Mohammadi et al., 2026), whereas continuation insufficiency is neither a migration score nor a count of missing facts but a failure of the current evaluator to distinguish accepted states with different conditional continuation profiles.

Benchmark design. A continuation benchmark should compare several states within one current-success class on shared mechanism-coupled and control futures. Multiple executors test portability; recovery should separate rollback, model repair, and known-action compatibility. Same-producer, repository-disjoint anchors limit producer and project shortcuts.