BENCHPULSE: MONITORING VERDICT AND RANKING DRIFT IN EVOLVING AGENT BENCHMARKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Agent benchmarks are increasingly maintained as software: tasks are repaired, tools evolve, verifiers are corrected, and leaderboard formulas change. Yet model scores from different benchmark versions are routinely compared as if the measurement contract were fixed. We introduce BENCHPULSE, a framework that models benchmark repository transitions as interventions on a five-layer evaluation contract—task, verifier, environment, execution policy, and aggregation. A 4,119-commit census across 15 agent benchmarks maps how these contracts evolve. CONTRACTIR represents the affected contract surface of a code change, while PULSESET uses cached execution dependencies and conclusion sensitivity to select a compact replay panel before observing new outcomes. Across 8,495 frozen development executions, real updates change 330 verdicts; a pre-specified held-out SWE-bench transition changes 17/642 additional frozen verdicts, and a separate 17-system audit changes 34 approximate 95% interval conclusions. A single verifier commit moves τ-bench by -5.66 percentage points, a τ^2-bench release moves individual model panels by up to 9.02 points and reverses 4 point-estimate orderings, and a BFCL aggregation change creates 893 strict point-order reversals among 5,886 model pairs, including 658 with at least a one-point gap under both contracts. Layer-aware monitoring re-evaluates 20.48% of cached executions and recovers 95.45% of changed verdicts; it recovers all observed full-panel point-order reversals with zero model calls. Two negative controls spanning 2,370 frozen executions produce no flips, showing that semantic-looking maintenance is not automatically consequential. These results establish benchmark evolution as a measurable source of reported-result drift and provide a practical continuous- evaluation protocol for versioned agent benchmarks.

1 INTRODUCTION

An agent benchmark result appears to be a compact fact: model A succeeds on 39.4% of tasks and model B on 37.4%. In practice, that fact is the output of a versioned program. A later commit can repair a task, change a tool's observable state, add a stopping condition, tighten a verifier, or rebalance a leaderboard. The same archived agent behavior can then support a different verdict—and a different reported model comparison. Benchmark maintenance is necessary; invisible changes in measurement are not.

Current longitudinal evaluations mainly refresh task content (White et al., 2025; Jain et al., 2025), evolve task difficulty (Guo et al., 2026), or rerun agents in controlled environment versions (Ishmam & Marino, 2026). Efficient-evaluation methods select a small static test set that preserves scores or rankings under one fixed contract (Yauney et al., 2026; Rubinstein et al., 2026). These lines establish that benchmark time and evaluation cost matter. The remaining measurement-side counterfactual is: *what changes when the evaluation program changes but the agent execution stays fixed?*

We answer this question with BENCHPULSE. The central insight is that a benchmark Git transition is an intervention on a structured contract, not generic software churn. CONTRACTIR normalizes each transition into task, verifier, environment, policy, and aggregation atoms. Given the changed atoms and a cache of old executions, PULSESET selects executions whose outcomes depend on the modified

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.

surface and prioritizes tasks capable of reversing close model comparisons. The new contract is then evaluated only on this sentinel panel. This produces a version-level report of changed verdicts, score displacement, ordering reversals, and comparability alerts.

The empirical signal is large. Replaying 1,980 public τ-bench trajectories through one historical verifier transition changes the pass rate from 59.95% to 54.29% and flips 112 verdicts. On τ^2-bench, a release-level contract transition changes 218 of 6,515 frozen executions and reverses 4 strict point orderings. Independently, applying two released BFCL aggregation contracts to the same 109 component-score profiles moves scores by 18.81 points on average and creates 893 strict point-order reversals. A pre-specified held-out SWE-bench verifier transition changes 17/642 verdicts across ten agent panels and moves a frozen panel by up to 3.08 points. These are changes in measurement, not new agent behavior.

Our contributions are:

- We formalize an agent benchmark as a five-layer, versioned evaluation contract and contribute a 4,119-commit, 15-repository history census plus CONTRACTIR, an auditable representation that maps code and data transitions to measurement semantics.
- We provide a multi-benchmark fixed-execution study of real agent-benchmark evolution across 1,980 τ-bench trajectories, 6,515 τ^2-bench executions, 642 held-out SWE-bench executions, 109 BFCL model profiles, and 2,370 negative-control executions. The study exposes score, verdict, and ranking changes attributable to exact releases and commits.
- We introduce PULSESET, a contract-conditioned monitoring policy. Across the two trace-level positive transitions it re-evaluates 1,740/8,495 cached executions and recovers 95.45% of all flips; layer-specific propagation recovers all observed full-panel point-order reversals without a model call.

2 EVALUATION CONTRACTS AND DRIFT

Five-layer contract. Let benchmark version v expose contract

$$C_v = (T_v, V_v, E_v, P_v, A_v), \tag{1}$$

where T defines tasks, references, and inclusion; V maps execution evidence to item outcomes; E defines tools, state, services, and dependencies; P defines budgets, retries, sampling, and stopping; and A aggregates item outcomes into scores, uncertainty statements, and model orderings. This decomposition captures executable agent benchmarks such as AndroidWorld (Rawles et al., 2025), τ-bench (Yao et al., 2024), τ^2-bench (Barres et al., 2025), and Terminal-Bench (Merrill et al., 2026) without treating every repository edit as a measurement change.

Fixed-execution intervention. An archived execution z contains a task identifier, trajectory, terminal state, and model/trial metadata. For versions whose changed predicates are evaluable on that record, $y_v(z)$ denotes its item verdict and $s_v(m)$ the aggregate score of model panel m. Comparing $y_v(z)$ and $y_{v'}(z)$ freezes agent behavior and attributes any difference to $\Delta C = C_{v'} - C_v$. For environment or policy changes that cannot be decided from cached state, the record instead identifies which sentinels require controlled re-execution. Our present causal replays use the former case and therefore make zero model calls.

We call a transition *consequential* when it changes an item verdict, an aggregate score, an uncertainty conclusion, or a strict model ordering. We report verdict flip rate, score displacement, Kendall's τ_b, strict pairwise reversals, and maximum rank movement. This definition keeps ordinary refactors and documentation edits outside the scientific object while retaining a genuine zero-effect semantic transition as a negative control.

3 BENCHPULSE

3.1 CONTRACTIR: CHANGES TO CONTRACT ATOMS

For every non-root first-parent commit, CONTRACTIR reads the complete structured diff, including additions, deletions, modifications, and renames. It extracts entities whose released behavior enters

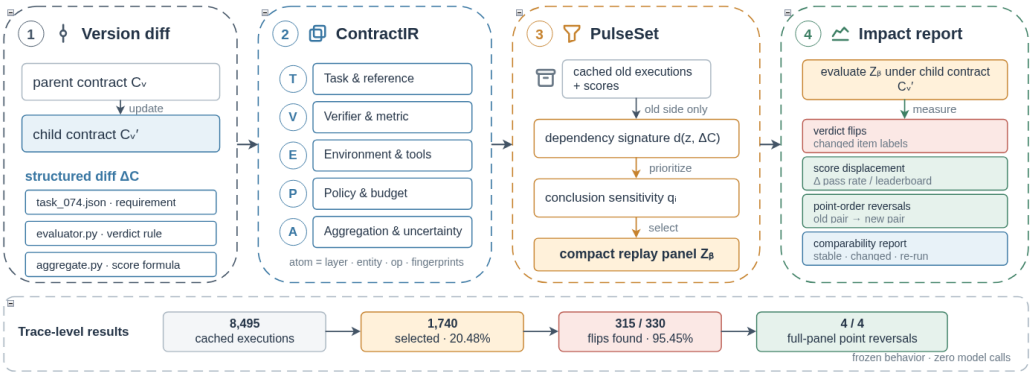

Figure 1: BENCHPULSE turns a benchmark transition into a version-level impact report. CONTRACTIR maps the structured diff to five contract layers; PULSESET joins the changed atoms with cached execution dependencies and old score margins, then evaluates a compact panel under the child contract. Across the two positive trace-level transitions (bottom), replaying 1,740/8,495 cached executions (20.48%) recovers 315/330 verdict flips (95.45%) and all four observed full-panel point-order reversals, with zero model calls.

C_v: task records and references; verifier functions and metrics; tool/environment modules and fixtures; run budgets and stopping configuration; and aggregation formulas. Each atom records its layer, operation, path, symbol or data entity, and parent/child fingerprint. Pure renames, formatting, tests without a released implementation change, and unrelated infrastructure remain explicit negative transitions.

The monitor processes every commit, rather than pre-filtering by message or known paths. Batch Git-object loading avoids one subprocess per file, while a rename fast path prevents large repository moves from becoming artificial semantic events. Missing objects in partial clones are isolated and logged. This design supports an auditable comparison against message keywords, semantic-path rules, their union, and raw churn.

3.2 PULSESET: CONCLUSION-SENSITIVE REPLAY

Let Z be cached old executions and b a replay budget. PULSESET receives only parent outcomes, parent artifacts, and ΔC; child outcomes are hidden until scoring. For each execution z, it builds a dependency signature $d(z, \Delta C)$ that indicates whether the trace, task, state, or score component can reach a changed atom. A transition-specific risk score $r(z)$ orders those dependencies. When a changed predicate is exactly decidable from cached state, this yields an exact dependency panel. Otherwise, it is a sentinel ranking.

Agent benchmarks are used to compare models, so flip recall alone is incomplete. For task i, we add conclusion sensitivity

$$q_i = \sum_{(a,b)\in\mathcal{P}} \frac{|R_{ia} - R_{ib}|}{|s_v(a) - s_v(b)| + \epsilon}, \tag{2}$$

where R_{im} is dependency-risk mass for model m, $\mathcal{P}$ contains non-tied old model pairs, and $\epsilon = 0.25$ percentage points stabilizes close ties. Executions are prioritized by dependency risk and their task's q_i. After replay, observed selected-row deltas are added to the old full-panel score; we never rerank models on a compositionally different subset.

Aggregation-only changes take a faster path. If A_v and $A_{v'}$ operate on cached component scores, BENCHPULSE propagates both formulas over the same vectors. No item trajectory needs replay, yet every score and ordering change is available exactly.

Table 1: Frozen replay datasets. "Units" are cached trajectories for verifier and release transitions and component-score profiles for aggregation.

Benchmark	Changed contract	Units	Panels	Primary endpoint
τ-bench	verifier	1,980	4 files	verdict flips
τ^2-bench	multi-layer release	6,515	17	flips + order
BFCL	aggregation	109	109 models	score + order
MLE-bench	aggregation/statistics	17	17 systems	CI conclusion
AppWorld	verifier normalization	1,728	3 agents	negative control
SWE-bench	XFAIL verifier	642	10 agents	held-out flips
SWE-bench	test-ID resolution	642	10 agents	negative control

4 EXPERIMENTAL DESIGN

Research questions. We ask: (RQ1) how frequently do released agent-benchmark contracts change, and which layers move? (RQ2) can real transitions change frozen verdicts, scores, or model orderings? (RQ3) how much replay does PULSESET need to recover those changes? (RQ4) what precision–recall trade-offs do message, path, churn, and five-layer contract signals provide for triggering consequence analysis?

Repository census. We freeze 4,119 non-root first-parent transitions from 15 public benchmark repositories spanning general, mobile, API, web, function-calling, desktop, software-engineering, interactive-tool, CLI, and tool-use agents. A prediction- blind, stratified sample of 120 complete patches is labeled by the authors using a written five-layer protocol. Two authors label disjoint halves; a third author independently duplicates 30 fixed-seed patches. Initial duplicate-label agreement is 83.3% (κ=0.664), with 73.3% exact layer-set agreement. Sampling balances the union path/message-positive and negative strata within repositories; population estimates use inverse inclusion weights for each repository–stratum cell. Low-confidence cases, duplicate-label disagreements, CONTRACTIR disagreements, and 20 fixed-seed agreements—84 patches in total—receive an independent author adjudication over the complete patch; this changes the binary label or layer set for 9 patches. We describe these as author annotations produced under the written protocol. Repository-cluster bootstrap intervals use 2,000 resamples.

Causal replay datasets. Table 1 summarizes four positive transitions, one secondary aggregation transition, and two negative controls. The τ-bench study applies the historical answer-only and answer- plus-database-state verifiers to public trajectories. The τ^2 study pairs 6,515 executions from 17 public panels across a complete banking-domain release; the old adapter exactly reproduces all 6,450 available non-infrastructure archived labels. The held-out SWE-bench transition was selected from a candidate order frozen before replay: the first direct-child verifier edit whose old and new semantics were executable on the full cached-log panel. BFCL holds 109 public component-score vectors fixed and applies the historical v3 and v4 formulas (Yan et al., 2024). AppWorld (Trivedi et al., 2024) replays a semantic-looking answer-normalization edit over 1,728 eligible executions as a negative control. A secondary MLE-bench aggregation audit compares 17 shared systems and the displayed mean/SEM cells across releases. Every adapter is pinned to source commits and has a clean-room deterministic rerun.

Monitoring baselines. We compare PULSESET with uniform random selection (1,000 fixed-seed resamples), old pass/failure or score-boundary selection, contract-coverage-only selection, path-only affected-task selection, and DISCO-style inter-model disagreement (Rubinstein et al., 2026) where aligned model–task panels exist. We report replay fraction, verdict-flip recall, score-shift error, rank-reversal recall, and rank-alert precision. The ranking endpoint is a strict point ordering computed against the original full panel; uncertainty-qualified comparisons are reported separately.

Table 2: Fixed-execution effects of real benchmark transitions. The agent output is unchanged wherever frozen executions or cached score vectors are available; MLE-bench is explicitly a released-statistics comparison. "Comparison changes" counts strict point-order reversals, except for MLE-bench's approximate 95% interval-conclusion changes.

Transition	Evidence mode	Verdict flips	Score shift	τ_b	Comparisons
τ verifier	fixed execution	112/1,980	-5.66 pp	–	–
τ^2 release	fixed execution	218/6,515	max +9.02 pp	0.925	4
SWE XFAIL	held-out fixed execution	17/642	max +3.08 pp	–	–
BFCL aggregation	cached score vectors	–	mean $\lvert\Delta\rvert$ 18.81 pp	0.696	893
MLE aggregation	released statistics	–	max 2.55 pp	–	34 CI
AppWorld norm.	fixed execution	0/1,728	0.00 pp	–	–
SWE ID resolution	fixed execution	0/642	0.00 pp	–	–

5 RESULTS

5.1 REAL CONTRACT TRANSITIONS CHANGE VERDICTS, SCORES, AND RANKINGS

τ-bench verifier change. The historical commit changes 112/1,980 verdicts and moves the aggregate pass rate from 59.95% to 54.29% (-5.66 points). Its scope is concentrated but substantial: among the 504 trajectories governed by the changed output contract, 22.22% flip. Because trajectories are frozen, this is a direct measurement effect.

τ^2 release change. Across 17 model panels, 218 executions move from failure to pass and none move downward. Seven panels shift by at least three points; the largest shift is +9.02 points. Four previously strict model pairs reverse point order, reducing Kendall's τ_b between old and new rankings to 0.925. All four retain at least a 0.5-point gap on both sides and 2 retain at least a one-point gap; paired task bootstrap intervals overlap zero for the four individual pair differences, separating point-estimate reversal from uncertainty-qualified reversal. A sensitivity analysis excluding the one public panel with a one-trial replay-versus-release- table discrepancy preserves 203 flips, the same +9.02-point maximum shift, and 3 strict reversals. Thus a valid release repair improves outcomes while changing reported historical model comparisons.

BFCL aggregation change. The same 109 component profiles differ by 18.81 points on average in absolute value under the two formulas, with a 34.42-point maximum. The change creates 893 strict point-order reversals among 5,886 pairwise comparisons; 658 remain reversals with at least one point of separation under both formulas and 495 with at least two points. One model moves by 60 positions. The reconstructed v4 score agrees with the released leaderboard within 0.0075 point, supporting exact formula recovery.

Held-out SWE-bench verifier change. The pre-specified XFAIL transition changes 17/642 verdicts across ten public agent panels and 2 unique tasks. The largest frozen-panel shift is +3.08 points. Because the candidate order, information boundary, and changed predicate were fixed before child labels were computed, this is a held-out contract-to-impact test rather than another transition selected for a large observed effect.

Measured impact is selective. The AppWorld normalization transition produces zero flips over 1,728 eligible executions. Independently, unique status-consistent resolution of truncated parameterized test IDs leaves all 642 frozen SWE-bench verdicts unchanged across ten public agents. These controls demonstrate that BENCHPULSE reports measured impact selectively: evaluator touches trigger analysis, while only observed outcome changes trigger comparability alerts.

Uncertainty conclusions also move. In the secondary MLE-bench audit, displayed means change by at most 2.55 points, but revised mean/SEM cells turn 34 pairwise comparisons from inconclusive to directional under the stated independent-normal 95% approximation. This extends the measured effect from point scores and orderings to uncertainty-qualified conclusions; strict point-order reversals remain a separately reported endpoint.

Table 3: Layer-aware monitoring with the evidence mode named explicitly. Replay fraction counts cached evaluation rows, not fresh model generations.

Benchmark	Mode	Replay	Flip recall	Reversal recall
τ-bench	exact cached dependency	5.66%	**100.00%**	–
τ^2-bench	non-oracle sentinel	24.99%	**93.12%**	**100%**
SWE-bench	held-out exact dependency	2.96%	**100.00%**	–
BFCL	exact score propagation	0%	–	**100%**

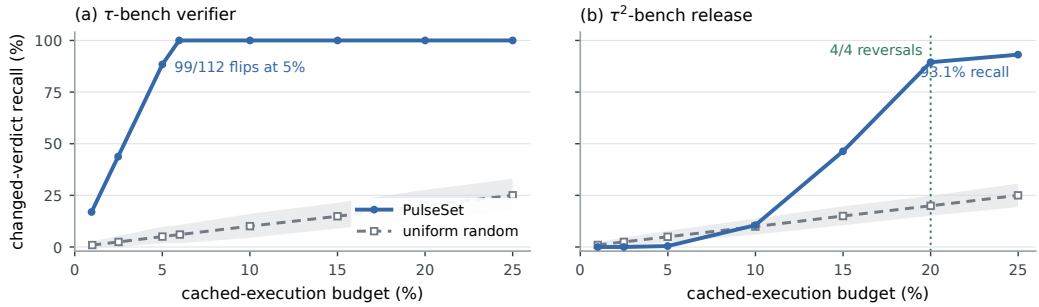

Figure 2: Early-budget verdict-flip recall from the saved selection curves. Lines show deterministic PULSESET selection and the mean of 1,000 uniform random samples; shading is the random baseline's empirical 95% interval. The exact τ-bench dependency panel reaches 100% recall at a 5.66% budget. On τ^2-bench, PULSESET recovers all four full-panel point-order reversals by 20% and reaches 93.12% flip recall at 24.99%.

5.2 LAYER-AWARE MONITORING PRESERVES THE CHANGE SIGNAL

Across both trace-level transitions, PULSESET re-evaluates 1,740/8,495 rows (20.48%) and recovers 315/330 changed verdicts (95.45%); Figure 2 shows how quickly this coverage emerges. The layer-specific BFCL path requires no trace replay and exposes all 893 pairwise reversals. All three analyses make zero model calls.

On τ-bench, the changed database predicate is decidable from cached terminal state. The exact dependency signature selects 112 rows (5.66%) and recovers every flip and the full score change. At the same budget, uniform random selection recovers 5.57% of flips on average. Even at a 5% budget, PULSESET finds 99/112 flips, versus 19 for contract coverage and 10 for old-pass selection.

On τ^2-bench, PULSESET selects 1,628 executions (24.99%) and finds 203/218 flips (93.12%), all four true full-panel reversals, and one false rank alert. Uniform random selection finds 25.01% of flips on average and all four reversals in only 0.9% of 1,000 resamples. On the stricter aligned 24-task panel, PULSESET finds three reversals and 78.44% of flips; DISCO-style disagreement finds one reversal and 21.10% of flips. Removing conclusion sensitivity and ranking only by contract coverage finds 56.42% of flips and 0 reversals. The complete cache scan and selection take 10.26 seconds on one CPU process; the paper-only 1,000-repeat audit takes 125.90 seconds total with 880 MB peak RSS. Contract-conditioned dependencies therefore complement, rather than duplicate, static disagreement selection.

The error anatomy in Figure 3 localizes the remaining gap: 201/203 captured flips satisfy the primary dependency-risk rule, whereas all 15 misses have no extra read, no dependency-risk flag, and no changed-document overlap. They cluster in five tasks but span 11 model panels. Every captured and missed flip touches a changed function, showing why broad call-graph reach alone does not prioritize the affected state-hash behavior. The one false rank alert is similarly localized: a 0.26-point partial crossing becomes an exact full-panel tie rather than a reversed ordering.

PULSESET's advantage concentrates at the operationally valuable early budgets. Once the aligned-task budget reaches 50%, both methods expose all four point-order reversals; coverage-only reaches

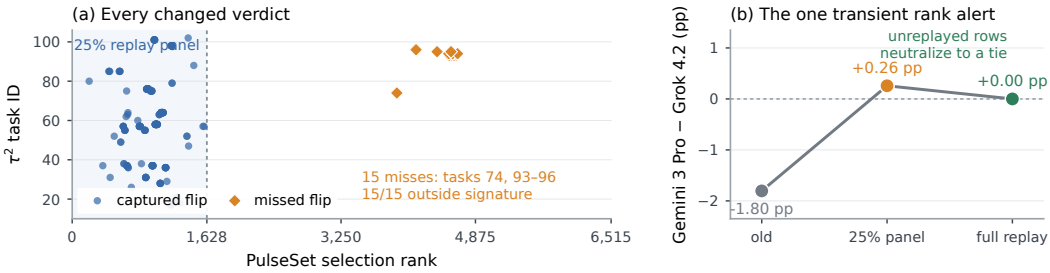

Figure 3: The τ^2 non-oracle sentinel's errors match its information boundary. Left: all 218 changed verdicts by deterministic selection rank; the 25% panel captures 203. The 15 misses lie in five tasks and all lack the extra-read dependency signature. Right: the single false rank alert is a near-tie—the selected panel moves the old gap from -1.80 to $+0.26$ points, and the unreplayed tail neutralizes it to an exact tie under the full contract.

92.20% flip recall and PULSESET 89.45%. This crossover identifies broad coverage as sufficient at high budgets and transition-conditioned prioritization as the stronger early monitor.

On held-out SWE-bench, the changed predicate depends only on whether a gold-expected test has cached status `XFAIL`. The resulting dependency signature selects 19/642 executions (2.96%) before child scoring and recovers all 17 verdict changes. This extends the complete CONTRACTIR-to-replay path beyond the two τ lineages without a model or judge call.

5.3 CONTRACT-HISTORY DETECTION

Contract-bearing maintenance is common rather than exceptional. The author-labeled audit contains 62 positive patches; inverse-inclusion weighting estimates that 68.72% of the 4,119 census transitions change at least one contract layer (95% repository-cluster interval 56.67–78.56%), or approximately 2,831 transitions. Environment and execution-policy changes are the most prevalent layers.

Table 4 shows that repository structure is already a strong trigger: semantic paths reach 83.47% F1 and the highest precision. The five-layer CONTRACTIR signal reaches 92.17% recall, 12.66 points above path, while assigning the changed surface needed by PULSESET. Raw churn reaches still higher recall but supplies no affected task, verifier, environment, policy, or aggregation entity. These AP values rank two-level trigger scores; CONTRACTIR is used for high-recall layer attribution rather than fine-grained transition ranking. These results motivate a two-stage monitor: inexpensive diff signals trigger an attributed consequence replay, and measured verdict or conclusion change—not the trigger alone—determines comparability. The two zero-effect controls in Table 2 demonstrate why this second stage matters.

6 RELATED WORK

Live and evolving benchmarks. LiveBench and LiveCodeBench refresh task composition to limit contamination and track aggregate performance over time (White et al., 2025; Jain et al., 2025). TRACE automatically evolves agent tasks while retaining reproducible trajectories (Guo et al., 2026); Code2Bench derives recent tasks from source repositories and validates them with executable tests (Zhang et al., 2026). BENCHPULSE is the comparability layer for this ecosystem: it follows changes to the measurement program itself and pairs the same execution across versions.

TimeWarp constructs six historical-style web versions and reruns agents to study capability robustness (Ishmam & Marino, 2026); OpenApps generates thousands of UI variations and finds large within-agent reliability swings (Ullrich et al., 2026). We freeze the execution and change real benchmark contracts. These designs answer complementary questions: whether an agent generalizes when the world changes, and whether an evaluation conclusion survives when its measuring program changes.

Process and evaluator audits. SeekBench defines groundedness, recovery, and calibration metrics over one fixed search-agent evaluation schema (Shao et al., 2026); AutoLibra induces fine-grained

Table 4: Population-weighted contract-change detection after independent author adjudication. Path provides the strongest F1; CONTRACTIR trades precision for 12.66 points more recall and explicit layer attribution.

Signal	Precision	Recall	F1	AP
Message	68.77	38.90	49.70	68.74
Path	**87.86**	79.50	**83.47**	**83.94**
Message ∪ path	77.58	82.85	80.13	76.06
Held-out churn	71.12	**94.72**	81.24	82.03
CONTRACTIR	71.14	92.17	80.30	70.95

trajectory metrics from open-ended feedback (Zhu et al., 2026). A concurrent tool-calling validity audit compares benchmark verdicts with expert trace judgments and documents snapshot-level evaluator disagreement (Vaghasiya et al., 2026). BENCHPULSE supplies the orthogonal longitudinal counterfactual: the trajectory stays fixed while the released task, verifier, environment, policy, or aggregation program changes.

Efficient and reliable evaluation. Micro-benchmarking asks whether a subset preserves fixed-contract scores or model orderings and finds random selection difficult to beat outside extreme reductions (Yauney et al., 2026). DISCO uses inter-model disagreement to condense static multiple-choice evaluation (Rubinstein et al., 2026). Sample Lottery uses behavioral volatility to select training instances (Xiao et al., 2026). PULSESET instead conditions on a specific contract transition and optimizes early detection of changed verdicts and conclusions. Our use of repeated random baselines and aligned model panels follows the strongest lessons from that literature.

Evaluator and statistical drift. Learned judges lose accuracy as generator responses and questions evolve (Singh et al., 2026). Bayesian and resampling methods improve uncertainty under a fixed scoring rule (Hariri et al., 2026). BENCHPULSE identifies a different variation source: deterministic task, verifier, environment, policy, and aggregation code can change the label attached to exactly the same execution. Sampling uncertainty and contract drift should therefore be reported together.

7 DEMONSTRATED SCOPE AND EXTENSIONS

The causal replay establishes consequences for verifier, release-bundle, and aggregation changes whenever the cached record contains the contract inputs. Environment transitions that alter unrecorded interaction state route to controlled sentinel re-execution, a dependency that CONTRACTIR identifies explicitly. Complete-patch labels, negative controls, commit-pinned adapters, and released per-transition evidence connect implementation changes to their measured consequence. Prediction-blind duplicate labeling and independent adjudication quantify the reliability of the author-annotated census, so prevalence estimates describe this audited repository population. The replay results describe the reported frozen panels of public model outputs; expanding the same protocol to additional historical submissions is a direct extension.

8 CONCLUSION

Agent benchmark versions are different measurement instruments. Across real repository transitions, the same agent behavior receives different verdicts, scores, and reported model orderings; one aggregation update alone reverses 893 of 5,886 point comparisons, including 658 separated by at least one point under both formulas. BENCHPULSE makes this evolution observable. CONTRACTIR identifies the changed contract surface, PULSESET replays the executions most capable of changing a conclusion, and layer-specific propagation eliminates unnecessary work. Re-evaluating 20.48% of cached trace rows recovers 95.45% of observed verdict changes and every observed full-panel point-order reversal on the development transitions; the pre-specified held-out SWE-bench replay recovers every additional verdict change from 2.96% of its frozen panel, without a model call. Continuous contract monitoring therefore makes benchmark improvement and longitudinal comparability compatible by construction.

AI USE STATEMENT

Generative AI tools assisted with topic discovery, hypothesis refinement, experimental-design feedback, method implementation, literature organization, result interpretation, and manuscript drafting and editing. They did not generate benchmark tasks, agent trajectories, model scores, replay outcomes, or the reported repository-patch labels. The authors performed all patch annotation and adjudication under the written protocol. Every literature claim was checked against the cited full text; generated code is covered by deterministic tests and clean-room reruns; all paper numbers are regenerated from saved JSON/CSV artifacts; and uncertain patch labels receive an independent author pass. The authors reviewed all AI-assisted work and take responsibility for the final text, claims, and artifacts. The review artifact is available at `https://anonymous.4open.science/r/benchpulse/`.

REFERENCES

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2-Bench: Evaluating conversational agents in a dual-control environment. *arXiv preprint arXiv:2506.07982*, 2025. URL https://arxiv.org/abs/2506.07982.

Dadi Guo, Tianyi Zhou, Dongrui Liu, Chen Qian, Qihan Ren, Shuai Shao, Zhiyuan Fan, Yi R. Fung, Kun Wang, Linfeng Zhang, and Jing Shao. Towards self-evolving agent benchmarks: Validatable agent trajectory via test-time exploration. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/da8860a2fe8ddb7589136853bcc313fc-Paper-Conference.pdf.

Mohsen Hariri, Amirhossein Samandar, Michael Hinczewski, and Vipin Chaudhary. Don't pass@k: A bayesian framework for large language model evaluation. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/f04edfc65463d020629673a4bc4c58e7-Paper-Conference.pdf.

Md Farhan Ishmam and Kenneth Marino. Timewarp: Evaluating web agents by revisiting the past. *arXiv preprint arXiv:2603.04949*, 2026. URL https://arxiv.org/abs/2603.04949.

Naman Jain, Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/94074dd5a072d28ff75a76dabed43767-Paper-Conference.pdf.

Mike Merrill, Alexander Shaw, Nicholas Carlini, et al. Terminal-bench: Benchmarking agents on hard, realistic tasks in command line interfaces. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/444a3737adaee10d86ad2ef5f74468e6-Paper-Conference.pdf.

Chris Rawles, Sarah Clinckemaillie, Yifan Chang, Jonathan Waltz, Gabrielle Lau, Marybeth Fair, Alice Li, William Bishop, Wei Li, Folawiyo Campbell-Ajala, Daniel Toyama, Robert Berry, Divya Tyamagundlu, Timothy Lillicrap, and Oriana Riva. Androidworld: A dynamic benchmarking environment for autonomous agents. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/01a83bc2f2732a58e6aa731e659e7101-Paper-Conference.pdf.

Alexander Rubinstein, Benjamin Raible, Martin Gubri, and Seong Joon Oh. DISCO: Diversifying sample condensation for efficient model evaluation. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/19c145aaad40927c51f4d10eaa339c20-Paper-Conference.pdf.

Jiaqi Shao, Yuxiang Lin, Munish P. Lohani, Yufeng Miao, and Bing Luo. Do llm agents know how to ground, recover, and assess? evaluating epistemic competence in information-seeking agents. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/f7c714fa40183e349891edf77ab5dcd4-Paper-Conference.pdf.

Janvijay Singh, Austin Xu, Yilun Zhou, Yefan Zhou, Dilek Hakkani-Tür, and Shafiq Joty. On the shelf life of fine-tuned llm-judges: Future-proofing, backward-compatibility, and question generalization. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/61b3a82a6a37574b64c6485e6a7e0535-Paper-Conference.pdf.

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*, 2024.

Karen Ullrich, Jingtong Su, Claudia Shi, Arjun Subramonian, Amir Bar, Ivan Evtimov, Nikolaos Tsilivis, Randall Balestriero, Julia Kempe, and Mark Ibrahim. Openapps: Simulating environment variations to measure ui agent reliability. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/bffc719468c22f9ba53e73696dd71858-Paper-Conference.pdf.

Jay Vaghasiya, Vishvesh Bhat, Muhammad Ahmed Mohsin, and Asad Aali. Benchmarking the benchmarks: A validity audit of tool-calling evaluation. *arXiv preprint arXiv:2607.02577*, 2026. URL https://arxiv.org/abs/2607.02577.

Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Benjamin Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Sreemanti Dey, Shubhi Agrawal, Sandeep Sandha, Siddartha Naidu, Chinmay Hegde, Yann LeCun, Tom Goldstein, Willie Neiswanger, and Micah Goldblum. Livebench: A challenging, contamination-limited llm benchmark. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/e4a46394ba5378b3f9a186a5b4c650d1-Paper-Conference.pdf.

Zhiping Xiao, Yusheng Zhao, Qixin Zhang, Jiaye Xie, Wanjia Zhao, Weizhi Zhang, Xiao Luo, Philip Yu, and Ming Zhang. Sample lottery: Unsupervised discovery of critical instances for llm reasoning. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/0f6cc80ad86e553d085842308e0fd2cb-Paper-Conference.pdf.

Fanjia Yan, Huanzhi Mao, Charlie Cheng-Jie Ji, Tianjun Zhang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. Berkeley function calling leaderboard. https://gorilla.cs.berkeley.edu/blogs/8_berkeley_function_calling_leaderboard.html, 2024.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*, 2024. URL https://arxiv.org/abs/2406.12045.

Gregory Yauney, Shahzaib Warraich, and Swabha Swayamdipta. How reliable is language model micro-benchmarking? In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/2e2960f2fe9e981f33f51c78656e3ca2-Paper-Conference.pdf.

Zhe Zhang, Runlin Liu, Aishan Liu, Xingyu Liu, Xiang Gao, and Hailong Sun. Code2bench: Scaling source and rigor for dynamic benchmark construction. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/f8e55d98b0c2569bd0aa25b076e6b3f8-Paper-Conference.pdf.

Hao Zhu, Phil Cuvin, Xinkai Yu, Charlotte Yan, Jason Zhang, and Diyi Yang. Autolibra: Agent metric induction from open-ended human feedback. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/20cb0454d5d9afca1c945d25e439659c-Paper-Conference.pdf.

A Reproducibility and Information Boundaries

Every reported number is generated from saved JSON/CSV artifacts. The main replays pin parent and child commits, source checksums, selection seeds, and adapter versions. PulseSet selectors are computed from parent outcomes, archived tasks/trajectories, and the parent/child contract diff; child labels, score shifts, and rankings are unavailable until scoring. Formal random baselines use 1,000 resamples with seed 1729. Clean-room reruns reproduce the τ, τ^2, and BFCL quantitative artifacts; checksums and commands are listed in the accompanying manifest.

B ContractIR Annotation Protocol

A patch is positive when it can change task inclusion or requirements, execution behavior, an item verdict, aggregate score or uncertainty, or model ordering. Positive labels name every applicable contract layer. Documentation-only, formatting-only, behavior-preserving refactors, test-only changes without a released implementation change, and unrelated infrastructure are negatives. The annotation packet omits all predictor outputs and sampling strata. Evaluation aborts on missing rows, duplicate identifiers, changed patch checksums, unresolved adjudication, or inconsistent labels.

C Additional Monitoring Controls

At the 5.66% τ-bench budget, the empirical 95% interval for uniform-random flip recall is 1.79–9.82%. At the 24.99% τ^2 budget, the corresponding interval is 19.27–30.73%. On the aligned task panel, random selection finds at least three of the four strict reversals in 7.9% of repeats; PulseSet finds three deterministically. These repeated controls distinguish transition-specific dependency coverage from generic small- set selection.