WHEN DOES SNAPSHOT EVALUATION PREDICT LIVE ROOT-CAUSE ANALYSIS?

Anonymous authors
Paper under double-blind review

ABSTRACT

Microservice root-cause analysis (RCA) agents are commonly evaluated on frozen incident snapshots because snapshot evaluation is reproducible and inexpensive. Deployed agents, however, query running systems whose telemetry evolves during diagnosis. It is therefore unclear whether capability estimates obtained from snapshots predict performance in live environments, or which properties of an incident make that extrapolation fail. We introduce a strictly paired evaluation protocol that instantiates each incident under three conditions: Snapshot, with frozen evidence; Live-fixed, with a live backend but the same observation cutoff as Snapshot; and Live, in which the agent starts no earlier than that cutoff and its observation window is not clamped. The protocol holds the fault specification, agent, tool interface, query budget, and scoring contract fixed across all three conditions, and the observation cutoff rule across Snapshot and Live-fixed; each arm still reads its own realization of the incident, so contrasts include realization variation. We further define three machine-checkable validity conditions: evidence sufficiency, execution-identity consistency, and scoring comparability. These conditions must hold before a paired result is attributable. Runs that violate them are classified as undiagnosable or incomparable rather than counted as model failures. On a cross-application benchmark spanning multiple fault mechanisms, the protocol measures Snapshot-to-Live transfer by paired score differences, strict-outcome agreement, and case-level flips, treating independent case families rather than repeated calls as the statistical unit. In an exploratory execution with three pre-registered models (deepseek-v4-flash, glm-5.3-flash, qwen3.8-27b) and two replicates per arm, only four cases, in two applications, had all three arms judged evidence-sufficient. On these, Snapshot $-$ Live-fixed pooled over models is $+9.0$ pp (strict-outcome agreement over usable replicate pairs 85.4%), descriptive only: four clusters fall below our pre-specified threshold of eight for reporting intervals, flips over usable replicate pairs run in both directions, and the direction is not established. The result supports no population, ranking or equivalence claim; a layer adding cases that failed the sufficiency check is reported apart and is not attributable. A complementary failure analysis demonstrates how omitted validity conditions can leave apparently successful evaluations without a defensible comparison. Together, the protocol and benchmark make the assumption that snapshot evaluation proxies live RCA an auditable empirical question.

1 INTRODUCTION

Snapshot evaluation of RCA agents is cheap: record an incident, freeze the evidence surface, run the agent offline. Live evaluation is expensive: the agent probes a running system and is scored against the outcome. Cost drives measurement choices, and snapshot benchmarks dominate (e.g., Xu et al., 2025). Whether the cheap measurement predicts the expensive one is the operative question.

Running both arms on the same incident does not license the comparison. A paired comparison is attributable only when three conditions hold: the evidence given suffices to diagnose the case; the artifact that executed is the one declared; and both arms are scored by one scorer at one revision. These conditions are not pipeline-specific — they are the conditions under which the estimand, the per-case paired difference in capability, is defined at all. Section 5 gives each a machine-checkable form.

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.

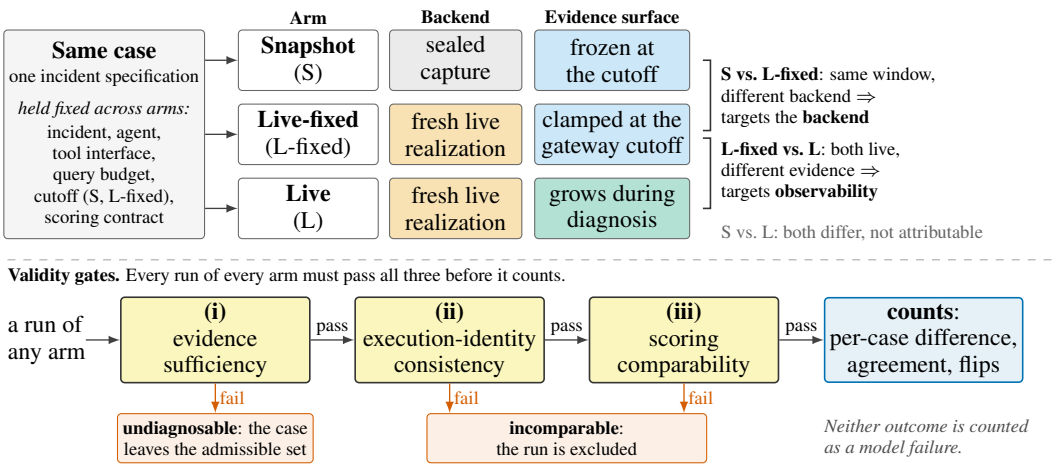


Figure 1: **The three-arm paired protocol.** One case is instantiated in three arms, holding the incident, agent, tool interface, query budget and scoring contract fixed across all three, and the observation cutoff across S and L-fixed: Snapshot (S) uses the case's sealed capture; Live-fixed (L-fixed) and Live (L) are each run on a fresh live realization of the same case, not on one shared realization. L-fixed is clamped at the gateway's own cutoff, which gives it the same observation window as S on its own realization; the protocol requires the gateway to record that cutoff in the run's audit, which it does not do at the revision measured in Appendix A.3.4, and a run without that record is undeterminable. L starts no earlier than the cutoff and its telemetry grows during diagnosis. S vs. L-fixed shares the observation window and differs in the backend, which it targets; L-fixed vs. L differs in the evidence surface and targets what additional observability contributes; S vs. L differs in both and is not attributable. Every contrast also spans two different realizations, whose variation pairing does not remove. A run of any arm counts only after it passes three validity gates: (i) evidence sufficiency, (ii) execution-identity consistency and (iii) scoring comparability. A case or run failing (i) is undiagnosable and a run failing (ii) or (iii) is incomparable; neither is counted as a model failure.

Two of the conditions can fail silently: in three deterministic reproductions, an integrity gate reports a pass while the executed artifact differs from the reported one; how often this happens in other pipelines is not estimated here.

Contributions:

1. A strictly paired three-arm evaluation protocol (Snapshot, Live-fixed, Live), that requires every held-fixed quantity to be recorded as an artifact (§4, Figure 1).
2. Three validity conditions — evidence sufficiency, execution-identity consistency, and scoring comparability — with machine-checkable definitions and measured status (§5).
3. An exploratory three-arm experiment with three pre-registered models; its main analysis covers only the four cases whose three arms were all judged evidence-sufficient, falls below our pre-specified reporting threshold of eight clusters, and supports no population, ranking or equivalence claim (§6).
4. A failure taxonomy mapping each discovered defect to the specific comparison it voids, using our own pipeline as primary evidence (§7).

2 Related Work

Our contribution is not a live RCA environment but the transfer question, a paired protocol and benchmark that make it measurable, and its validity conditions (more in Appendix B.1).

Snapshot RCA over production telemetry is established. Xu et al. (2025) evaluates LLM agents on frozen captures of real incidents and scores them deterministically on root component, root reason

and root-cause time. Our snapshot arm is inside this paradigm deliberately. We do not claim a better snapshot benchmark; we ask whether such an estimate predicts what the same agent does against a running system, which a snapshot benchmark cannot determine. Gopal & Krishnan (2026) split failures, after scoring, into evidence present but unused and evidence absent; condition (i) is checked before scoring and decides whether a case enters a paired comparison at all.

Live construction and snapshot replay already coexist in one benchmark. Wang et al. (2026b) builds cases by runtime verification and exposes a snapshot-interactive replay, motivated by run-to-run variation in live evidence, so the co-presence of the two evidence regimes is prior art. For web agents, a rigorous online assessment suggests over-optimism in previously reported results (Xue et al., 2025). The distinction we draw is not existence but pairing with measured comparability: we hold case, budget, harness and rubric fixed across arms and treat the arms' comparability as a measured object rather than an assumption. §5.2 shows what happens when it is absent.

Live realism is not a defensible novelty claim. Clark et al. (2026) provides a live environment across multiple applications with a broad fault catalogue and compound failure modes (see also Chen et al., 2025; Gong et al., 2026). We claim no credit for a live environment, fault injection or tool-using agents. Here realism is a controlled variable—the purpose of the evidence-restricted arm L-fixed (§4.3), executed in §6.

Integrity signals that pass while the property they certify does not hold are a known failure shape. Wang et al. (2026a) is our nearest methodological sibling. We cite it as precedent for that shape, not as a result we extend. Our instance sits in a different substrate — divergence in the execution path — shown by deterministic reproduction within our pipeline.

Attestation of what ran is solved work with a deployed design. Torres-Arias et al. (2019) specifies supply-chain attestation for whether the declared artifact is the executed one. We do not re-derive it. What we report is narrower: a pipeline carrying its own integrity gate still admits the divergence such designs prevent, within a stated threat model, trusted-component list and proof boundary.

3 Problem Formulation: the transfer question and its estimand

3.1 The shared object and the unit of analysis

An *incident specification* σ is a sealed description of one fault: a public evidence boundary (task, topology, result schema, frozen logs, metrics), a private ground truth, and a reproduction descriptor for a running system. All arms instantiate the same σ; none sees the private side. Write $\mathcal{B}$ for base cases; $\mathrm{base}(\cdot)$ maps any run or variant to its base case.

> **Definition 1 (unit of analysis).** The paired unit is the base case $b \in \mathcal{B}$; cases in the same application–mechanism family form one statistical cluster. Variants, re-recorded revisions, replicates and frozen recordings are *never* independent units.

Repeated calls do not increase the number of cases or clusters. Case scores are aggregated within application–mechanism clusters before cross-cluster averaging (§6).

Arms.

$$\mathsf{S} = (E_{\mathrm{frozen}}, V_{\mathrm{cutoff}}), \qquad \mathsf{L} = (E_{\mathrm{live}}, V_{\mathrm{live}}), \qquad \mathsf{L\text{-}fixed} = (E_{\mathrm{live}}, V_{\mathrm{cutoff}}).$$

$\mathsf{S} \leftrightarrow \mathsf{L\text{-}fixed}$ is the *attributable* contrast (same observation window, different backend and realization); $\mathsf{L\text{-}fixed} \leftrightarrow \mathsf{L}$ targets what extra observability buys; $\mathsf{S} \leftrightarrow \mathsf{L}$ varies both and is reported only as non-attributable. L-fixed is specified in §4.3 and executed in §6.

The pairing relation.

> **Definition 2 (pair).** Runs $r = (a, m, u, j)$ and $r' = (a', m', u', j')$ are *paired*, $\pi(r, r')$, iff $a \neq a'$, $\mathrm{base}(u) = \mathrm{base}(u')$, and model, decoding settings, tool budget and scoring rubric coincide.

Pairing is at the base case: R replicates of one case give R paired runs and **one** paired unit; scores average within case before any cross-case comparison.

3.2 THE ESTIMAND

Let $y_a(b)$ be arm a's mean score on b over the diagnosis sub-rubric common to both schemas and $c_a(b) \in \{0, 1\}$ its case-level strict-correct outcome. The paired score difference Δ and the strict-outcome agreement α are

$$\Delta_{a,a'}(b) = y_a(b) - y_{a'}(b), \qquad \alpha_{a,a'}(b) = \mathbb{1}[c_a(b) = c_{a'}(b)] \,.$$

Definition 3 (estimand). $\theta_{a,a'} = \big(\mathbb{E}[\Delta_{a,a'}(b)],\ \mathbb{E}[\alpha_{a,a'}(b)]\big)$, reported *jointly*, broken out by fault mechanism, with the full per-case table of case-level flips.

The primary θ takes $(a, a') = (\mathsf{S}, \mathsf{L}\text{-fixed})$, sign fixed so that positive means snapshot scores above a live agent with the same observation cutoff. Estimates per layer: Table 2.

When the estimand is undefined. θ is **undefined**, not merely imprecise, on any base case where a validity condition fails: the evidence does not suffice for diagnosis, the artifact that executed differs from the one declared, or the arms' scores came from different scorers. Then $y_a(b)$ does not measure the quantity θ averages; the case leaves the admissible set, recorded in its own row, never imputed. Conditions: §5; omissions: §7.

The protocol's reporting rule. The protocol specifies reporting the per-case paired difference (Miller, 2024), the case-level success and failure changes, and intervals clustered at the application $\times$ mechanism level together, never using a non-significant result to argue equivalence. Intervals are withheld below N = 8 independent clusters with both arms, and points below it are labelled descriptive. As P6 replicate pairing is incomplete, §6 reports flips and agreement over usable replicate pairs as descriptive diagnostics, not this case-level estimand; replicate pairs are not independent cases and add no effective sample size.

4 A STRICTLY PAIRED SNAPSHOT–LIVE PROTOCOL

4.1 ONE INCIDENT SPECIFICATION, INSTANTIATED PER ARM

Each sealed case instantiates one incident specification (§3.1): the snapshot arm uses the case's sealed capture; L-fixed and L are each run on a fresh live realization of the same case. Case and expected answer are one object by construction. The protocol requires each held-fixed quantity to be **recorded as an artifact**, making the pairing auditable; at the revision measured in Appendix A.3.4, the gateway's cutoff for L-fixed is not. No gate verifies cross-arm identity; it is readable but unenforced (§5.3). Contract and scope: Appendices A.1 and B.2.

4.2 THE INTENDED DIFFERENCES, AND FOUR DIFFERENCES PAIRING DOES NOT REMOVE

Across arms, the backend and the evidence surface — frozen or live — are the **intended** differences (§3.1). Three further differences are measured rather than assumed away; a fourth, realization, follows from the design.

(i) Environment existence. The live arm populates recovery and teardown fields from container subprocesses and live probes; the snapshot arm has no environment and populates them **vacuously** — `accepted: true` in all cases, an **auto-true analogue** rather than an absent field. An absent field fails loudly; an auto-true analogue fails silently. The paired outcome is therefore on the diagnosis sub-rubric common to both schemas; numbers over the full live rubric are inadmissible.

(ii) Freshness dynamics. The live arm's surface grows during investigation; the snapshot arm is fixed at the cutoff. Pairing aligns the observation cutoff, not the evidence each realization produces before it, nor whether the surface keeps growing.

(iii) Scorer implementation in the pilot. Ninety of one hundred rubric weight is computed identically on both arms; the remaining ten uses different predicates (§5.3). P6 requires a common scorer.

(iv) Realization. S reads the case's sealed capture, while L-fixed and L each run on a fresh live realization of the case, so every contrast spans two realizations; pairing does not remove the variation between them.

4.3 THE EVIDENCE-RESTRICTED LIVE ARM, AS SPECIFIED

We specify a third arm, L-fixed: the environment exactly as in L, the observation window restricted to the snapshot's cutoff by clamping at the gateway's own cutoff; the protocol requires the gateway to record that cutoff in the run's audit, which it does not do at the revision measured in Appendix A.3.4, and a run without that record is undeterminable. Contrasts as in §3.1. Sign: $S - L$-fixed, positive pre-registered. The pre-registration was amended before any model run of the three-arm experiment; Appendix F lists each deviation. All three arms were planned on nine units in eight independent clusters across four applications, exactly at the floor of 8 (zero margin); admissibility review admitted two, and the executed record is in §6.

Status. Executed in P6 (§6), in every origin run that started, before L in each block. As every origin S intent is refused by design, the origin runs completed 14 (r15), 11 (x2), 18 (r18u), 21 (r15u) and 8 (x2u) pre-registered-model runs of L-fixed and L together, not split by arm in the record (Table 9). Before P6, the one audited run of its case driver recorded an empty observation cutoff and a zero-byte tool audit. Each L-fixed run is judged by the three conditions; one without the gateway's cutoff record is undeterminable. Reproduction requires a container runtime and host-bound attestation (Appendix A.1). An $S \leftrightarrow L$ statement without its L-fixed pair still carries the environment-existence confound of §4.2(i).

What the protocol fixes before any score is read. A run from a checkout with uncommitted changes is incomparable, not rescued. The original 26-case plan recorded $N_{\text{eff}} = 14$ in error; its listed partition has 17 clusters (Appendix A.1.1). Neither count is the executed P6 sample size.

5 VALIDITY CONDITIONS FOR AN ATTRIBUTABLE SCORE

Pairing requires **(i) evidence sufficiency**, **(ii) execution-identity consistency** and **(iii) scoring comparability**. The measurements below concern the earlier two-case pilot and pipeline audits, not P6. They motivate P6's gates (§6); their effects on scores or rankings are not measured.

5.1 CONDITION (I) — EVIDENCE SUFFICIENCY

Definition: correct diagnosis derivable from the evidence surface alone. *Check:* acceptance block plus a constructive full-marks proof. *Silent failure:* an undiagnosable case scored as a diagnostic failure of the agent.

Measured: passes on both arms of both paired cases — on *these two cases*. **We have no measured instance of this condition failing silently.** Where evidence sufficiency failed it failed loudly: four of eight certification records carry a named evidence-gate failure class and are marked ineligible for capture as evaluation data — eight certification records, not a paired sample size.

5.2 CONDITION (II) — EXECUTION-IDENTITY CONSISTENCY

Definition: the bytes that executed are the bytes the record declares. *Check:* declared pin versus digest of what ran. *Silent failure:* gate passes on a well-formed record whose revision does not determine the code.

Three deterministic reproductions: a substituted supporting module executes while the pin leg reports a match; poisoned bytecode executes while the source path digests to the pin of the sealed case package (the stored form of a sealed case, §4.1); a directory with no seal artifact reads as sealed and accepted, no seal document opened — the sealing flow ran and its refusals fired, so no existing capture is fabricated and no published figure changes; the defect is that certification would not detect a bypass.

Within a stated threat model, a pin stored in the same repository cannot establish the entry point's own identity; out-of-process attestation is solved work, acknowledged and not re-derived (Torres-Arias et al., 2019). We established cross-session tree-state identity; the unavailability is specific to the entry point. Threat-model and trust-boundary detail is in Appendix A.2.1.

On the pilot data: three of four snapshot runs record a dirty worktree admitted by a flag that permits uncommitted changes, so the declared revision does not identify the bytes that executed and no post-hoc attestation can repair them. Details are in Appendix C.1.

5.3 Condition (iii) — scoring comparability

Definition: both arms scored by the same scorer at the same revision. *Check:* compare producer revisions and the rubric executed. *Silent failure:* two well-formed score files, identical check names and weights, difference read as an effect of the arms.

Measured, both halves together: ninety of one hundred rubric weight (Appendix A.2.5) is computed by numerically identical rules with identical thresholds on both arms; the remaining ten is computed by two different predicates — time-windowed on the live side, untimed local on the snapshot side, expected labels from different sources. **Same rubric: PASS. Same scorer: UNDETERMINED.**

A unified scorer exists, certified in all eight certifications and not used by the eight paired-pilot runs. Rescoring failed because the weight-ten check — the divergence needing resolution — cannot be evaluated without the missing evidence contract; full rescoring detail is in Appendix A.2.3.

The stop rule. The live arm records its score producer's revision; the snapshot arm records none. **No cross-arm number may be stated until that closes**; it gates the headline agreement, the restricted-surface and attributable comparisons and the agent-fixed residual on the pilot record (Appendix C.4). In the three-arm experiment, scoring every arm by the same scorer at the same revision is a hard gate (Appendix F.2). Closing it means scoring both arms at one scorer revision; the cost is re-executing the pair under a consistent time contract, not local recomputation.

A lineage reconstruction for one pilot case is a finding, not a certification: the pre-declared check still cannot be run and the count passing (iii) remains zero (Appendix C.2).

5.4 Which failures are silent, and why the distinction is load-bearing

In the pilot pipeline, condition (i) has a gate with a named refusal. Conditions (ii) and (iii) lack effective enforcement: the identity gate passed on all three reproductions, dirty runs emitted complete records, and scorer revisions were not compared. Silent failures leave well-formed numbers; explicit exclusions remain visible but can still change the evaluated population.

Table 1: Validity conditions: measurements on the pilot and pipeline audits, not P6.

Condition	Definition	Check	Silent-failure signature	Measured status
(i) Evidence sufficiency	Correct diagnosis derivable from the surface alone	Acceptance block + constructive full-marks proof	**None — fails loudly**: named failure class, ineligibility flag	**PASS** both arms of both cases; loud failures elsewhere
(ii) Execution-identity consistency	Bytes that executed are the bytes declared	Declared pin vs. digest of what ran	Gate passes; record well-formed; revision does not determine the code	**FAIL silently**: three reproductions; three of four snapshot runs dirty
(iii) Scoring comparability	Both arms scored by one scorer at one revision	Producer revisions; checks, weights, predicates	Two well-formed score files; nothing compares them	**FAIL**: 90/100 comparable; weight-10 differs; no snapshot producer revision

6 Paired Three-Arm Results

What was executed. The protocol ran as experiment P6 (every run and departure: Appendix F). Of nine planned units (eight clusters, four applications), admissibility review (P4) admitted two (case-0602 and 0605, wave r15); it admitted none of three re-captured units (r18), none of one Wave-A

unit (x1) and two of three Wave-B units (case-0501 and 0613, x2). The primary tier thus holds four units in two applications (online-boutique, r15; istio-bookinfo, x2). Tier U, analysed apart, holds eight units P4 did not admit: three each in r15u and r18u, one in x2u, and one in x1u, blocked before any P6 run. The primary population is the units whose three arms P4 judged evidence-sufficient, not RCA faults in general. Each unit was planned with S, L-fixed and L for three pre-registered models, two replicates per arm (models 1–3 are the provider model IDs `deepseek-v4-flash`, `glm-5.3-flash` and `qwen3.8-27b`, called through one OpenAI-compatible endpoint by one shared harness, with at most 8 turns, 12 observation calls and a 1600-token output cap in every arm). A unit is one sealed case; a cluster is the application–mechanism family, the statistical unit; a replicate is one repetition of a unit's arms. Scores are rubric partial scores in percentage points; the strict outcome is success or failure. Pairs with an undeterminable or incomparable run are dropped, never imputed or rerun; S ran in a separate local section. Two exploratory models are described per wave in Table 7, never pooled; run counts are in Table 9.

Waves, layers and the reporting rule. The joint analysis ran once (per-wave status in §8) on pre-declared layers (Appendix E): J1 as pre-registered (r15), J3 over all admitted units, and J5, the admitted units plus tier U. A layer carries an interval only if at least 8 clusters have all three arms (a threshold we pre-specified, not a general statistical rule) (J3s families: 2); otherwise its points are descriptive, marked (d) in Table 2, and flips and agreement over usable replicate pairs are descriptive diagnostics, not the case-level estimand of §3.2. Status: J1 below floor (descriptive only); J3 below floor (descriptive only), 4 clusters; J5 reported, 8 clusters.

Table 2: Joint P6 layers: clusters with all three arms, and each contrast pooled over models (estimate [95% cluster-bootstrap interval], percentage points); Cases: pooled primary cases. (d): descriptive, no interval: fewer than 8 clusters have all three arms. n/a: no estimate, or the family is not reported. Each contrast is pooled over its own usable replicate pairs (a pair drops when either run is undeterminable or incomparable), so the three columns are not an algebraic decomposition: S − L need not equal (S − L-fixed) − (L − L-fixed). Usable replicate pairs (S − L-fixed / L − L-fixed / S − L): J1 10/10/11, J3 18/16/19, J5 53/53/56. Models are averaged within case, cases within cluster, and clusters weigh equally. The floor of 8 clusters is our pre-specified reporting threshold, not a general statistical rule.

Layer	Clusters	Cases	S − L-fixed	L − L-fixed	S − L
J1 (r15)	2	2	+5.6 (d)	+19.4 (d)	−5.6 (d)
J2	2	2	+5.6 (d)	+19.4 (d)	−5.6 (d)
J2s	0	0	n/a	n/a	n/a
J3	4	4	+9.0 (d)	+1.4 (d)	+5.6 (d)
J3s formal9	2	2	+5.6 (d)	+19.4 (d)	−5.6 (d)
J3s extension	2	2	+12.5 (d)	−16.7 (d)	+16.7 (d)
J5 (all waves)	8	11	+11.1 [+4.9, +17.5]	+5.9 [−3.8, +16.0]	+3.8 [−13.2, +17.4]
J5 S=insufficient, L-fixed=insufficient, L=insufficient	1	1	n/a	n/a	n/a
J5 S=insufficient, L-fixed=insufficient, L=sufficient	5	6	+15.3 (d)	+9.4 (d)	+2.8 (d)
J5 S=sufficient, L-fixed=sufficient, L=sufficient	4	4	+9.0 (d)	+1.4 (d)	+5.6 (d)

The primary contrast and the hypotheses. S − L-fixed holds the fault specification and the observation cutoff rule fixed, but S and L-fixed read different realizations (§4.2), so it also carries realization variation. Pooled over the three models, it is +5.6 pp on J1 (*n/a (below floor)*; agreement 91.7%; 2 cases, 2 clusters) and +9.0 pp on J3 (*n/a (below floor)*; agreement 85.4%; 4 cases, 4 clusters). On J3, L − L-fixed is +1.4 pp and the non-attributable S − L is +5.6 pp; the three contrasts are pooled over different usable replicate pairs (18, 16 and 19 of 24 executed), so they do not decompose. A post-hoc check, not pre-registered, on the 9 of 12 model–case cells present in all three J3 contrasts gives +9.0, +1.4 and +7.6 pp, which does decompose; on J5 the pair sets also differ within cells, and we did not reconstruct them. The mean partial-score difference and strict-outcome agreement answer different questions. S scores higher than L-fixed on both layers, but intervals are withheld; the strict-outcome flips (S→F/F→S, models 1–3, over usable replicate pairs) are 0/0, 0/0 and 0/1 on J1, and 2/0, 0/0 and 0/1 on J3. These four units support no population estimate, ranking transfer or equivalence claim. Reported agreement rates are aggregated summaries, not raw agreement counts divided by total usable pairs. Table 6 (Appendix E) gives per model the flips and the verdicts on the pre-registered hypotheses: H1, S and L-fixed agree on a majority of strict outcomes (agreement is computed over usable replicate pairs; falsified by agreement at or below half, or by a clustered interval on the paired difference that excludes zero); H2, S − L-fixed is positive (falsified by a negative point or an interval covering zero); H3, L is at least L-fixed (falsified by L

below L-fixed on the point). The H2 rule counts an interval covering zero as falsifying; we keep the recorded verdicts but read such an interval only as insufficient evidence for the direction, not as evidence against a positive effect. The interval clause in the recorded H1 rule concerns partial-score differences, not majority strict-outcome agreement; we retain its verdicts only as protocol history and do not treat that clause as evidence about H1. Every H1–H3 verdict, for each model on J1 and on J3, is `not_evaluated`, because both layers are below floor (descriptive only): the hypotheses are not testable here, and the models do not disagree on any verdict.

Tier U. J5 pools the primary units with tier U, whose units P4 did not find sufficient in every arm, so its contrasts are not attributable in the sense of Section 5. Being reported with 8 clusters, its pooled S − L-fixed ($+11.1$ pp, 11 cases; strata in Table 2) points the same way as J3 ($+9.0$ pp) and carries the interval $[+4.9, +17.5]$ pp; this exploratory interval does not back the primary result: its all-sufficient stratum is the J3 units themselves, so the agreement is not independent, and the units it adds have an insufficient cell, so neither its point nor its interval is attributable to evidence timing; J5 is never pooled into J1–J3. Its per-model verdicts (Table 6; Appendix E) describe the pooled tier and are not attributable to evidence timing.

7 Failure Analysis: Which Condition Voids Which Comparison

The three-outcome separation. The protocol distinguishes undiagnosable cases, incomparable runs and model errors (Kim et al., 2026; Lu et al., 2026). The pilot has no undiagnosable case, but three snapshot runs fail identity checks; none of its pairs passes all three conditions (Appendix D). Its outcomes cannot support a cross-arm effect estimate.

7.1 Why the remedy is external attestation rather than better discipline

All specimens come from the pipeline used for this study. Our three-leg integrity check reported identical for two trees differing in exactly the state this paper reports: a planted bytecode file left all three legs unchanged and the untracked count at one hundred twenty-six before and after. No published error was caught by an auditor reasoning more carefully; every one was caught by an independent audit that measured. Further instrument defects in our own pipeline, including how the `L-fixed` restriction is recorded, are in Appendix C.5.

7.2 Three named classes, each with a remedy

Count coincidence. Four cardinality coincidences across unrelated populations in one repository. *Verify sets by difference in both directions, never by count.*

Bounded or greedy match. Redefines the population a count covers — four instances in one round, two per audit pass. *Assert a canary member present; cross-check from a differently built extractor.*

Instrument unable to answer. A loader failure yields a failure verdict, so a two-leg criterion passes while the instrument is broken. *Third leg classifies loader failures and empty selectors as instrument-broken.*

Three further classes appear in Table 3: re-execution revision, invocation not recorded and threshold mismatch; detail in Appendix A.3. Three admissibility screens join them, each voiding condition (i) by being unanswerable rather than by failing: no reachability predicate, an absent gate input, and an acceptance outranking its own failing sub-check.

8 Limitations and Scope

These findings are a reproducible case study in one framework and development history, not an estimate of how common such defects are. In P6 only 4 (J3) and 2 (J1) clusters have all three arms, below the floor of 8, so the primary layers are descriptive only; J5, which pools the admitted units with tier U, reaches the floor but is not attributable (§6). The pilot is one base case (App. D).

The executed experiment departs from the pre-registration in ways Appendix F lists in full. The snapshot stage over twenty-six cases was not run. Admissibility review admitted only two of the

Table 3: Failure classes, the condition each violates, the comparison it voids, and whether it is detected.

Failure class	Cond.	Comparison voided	Detected?
Substituted module executes; pin leg reports match	(ii)	Any pairing using that run	No
Worktree dirty; revision does not identify bytes	(ii)	Those snapshot runs as units	Recorded, not refused
Re-execution revision alters model-visible option set	(ii)	Re-run pair and stored arm	No
Invocation not recorded	(ii)	Attribution of state to a cause	No
Two scorers; snapshot arm lacks producer revision	(iii)	Every cross-arm number	No
Executed threshold departs from declared one	(iii)	Declared-rubric readings	No
Identity leg skipped when both pin halves absent	(ii)	Leaves no negative answer	No
Integrity check reports identical trees that differ in that state	meta	The gate as evidence	No
Reachability screen has no implementation	(i)	Reachability half of the admissibility screen, every arm	Not implemented
Gate input absent; gate raises before any logic	(i)	Evidence-sufficiency gate for thirteen of twenty-six	No
False sub-check rolls up into a true acceptance	(i)	The sealed package's own certification	By a separate gate
Directory total standing in for a unit count	unit	Apparent N and every interval	No — count well-formed

nine planned units. Six further amendments, all signed before any score was read, re-captured units, added extension units and created a separately analysed tier of non-admissible units. The origin S runs were refused by design, and the S arm of record ran in a separate section. Truncation moved to 2026-09-26T05:00Z. The primary tier covers fewer than eight clusters. Its results are therefore descriptive, and intervals appear only for a layer that reaches the floor. Waves were executed separately; at the one joint analysis their status was: r15 (primary): analysed; r18 (primary): no_units; x1 (primary): no_units; x2 (primary): analysed; r15u (U): analysed; r18u (U): analysed; x1u (U): not_started; x2u (U): analysed. Further costs (App. F.8): selection on pre-model verdicts, revisions confounded with application and mechanism, and S run apart from the block order (three-second capture offset). Missingness can change each contrast's case/model composition; complete-pair analysis does not establish absence of selection bias.

L-fixed ran in every origin run that started (§4.3), but its cutoff record is judged run by run, and an S↔L statement without an L-fixed pair carries the environment-existence confound of §4.2. Of three evidence levels, only a reproduced check failure (an invalid or identity-unclear artifact passes a check) is established, three times; a measured effect of a failed condition on inputs, samples or scores, or on a ranking or the Snapshot↔Live conclusion, is not: §6 measures contrasts on runs that pass the conditions.

9 Conclusion

Whether snapshot evaluation predicts live RCA is measurable with a strictly paired protocol if three machine-checkable conditions hold (§5). Integrity gates can pass while the executed artifact differs from the reported one (three deterministic reproductions); Table 3 maps twelve failure classes to the comparisons they void. On the four admitted units (§6), S − L-fixed on J3 is $+9.0$ pp pooled over models, below our pre-specified reporting threshold, with flips over usable replicate pairs in both directions: descriptive only, direction unestablished. J5 adds units not sufficient in every arm and is not attributable.

AI USE STATEMENT

In this work, we used LLM-based coding agents extensively, including for tasks whose disclosure is required. They proposed and revised hypotheses: they drafted the pre-registration, including its hypotheses, as well as its later corrections and amendments. They proposed and revised the paired protocol, its estimand and its validity checks, and made analysis decisions, including adjudicating corrections to the pre-registration and decisions on how it applies. They implemented and tested the benchmark tooling and the experiment harness, rebuilt the benchmark cases used by the three-arm experiment, orchestrated and audited the execution of experiment runs, and interpreted pilot measurements. They also executed the three-arm experiment (P6) runs, ran its analysis and drafted the interpretation of its results; the authors did not select results. We further used them to identify related work, to draft and revise the text of this paper, and to check references against their original sources. The paper states no theorems and uses no human-subject data, so disclosure tasks concerning mathematical proofs, surveys, interviews or qualitative data are not applicable. The authors fixed the research question, set the budget and approved the first amendment to the pre-registration; they delegated the decisions of the second amendment, other than its budget, to the agents, which recorded them before the runs they govern. The agents also made later execution decisions, such as the replicate-major run order and the choice of exploratory models, and recorded each before the runs it governs. The authors answered the agents' questions at decision points and decided a correction the agents escalated to them; the other corrections and analysis decisions were adjudicated by the agents. Agent-produced code was reviewed by separate agent instances and checked by test gates; neither step caught every defect, and Sections 5 and 7 report defects in our own pipeline, including runs whose executed code could not be identified. We take full responsibility for the final content of this work, including text, claims and artifacts produced with the aid of LLMs, and for the analysis decisions the agents made. The LLM-based RCA agents that the protocol evaluates are the objects of study rather than research assistance, and are outside the scope of this statement.

REPRODUCIBILITY STATEMENT

The estimand and statistical unit are defined in Section 3; the paired protocol is in Section 4. P6's models and budgets are in Section 6, with execution rules and deviations in Appendix F. Appendix A.1 records the earlier pilot's contract. Section 5 defines the three validity checks; Appendix A.2 gives the threat model, trust boundary and pilot rescoring details. This PDF includes the appendices, but no separate code or data supplement accompanies the submission. We plan to release the benchmark tooling, experiment harness and validity checks publicly.

REFERENCES

Reem Aleithan, Haoran Xue, Mohammad Mahdi Mohajer, Elijah Nnorom, Gias Uddin, and Song Wang. SWE-Bench+: Enhanced coding benchmark for LLMs. *arXiv preprint arXiv:2410.06992*, 2024. URL https://arxiv.org/abs/2410.06992.

Ibragim Badertdinov, Alexander Golubev, Maksim Nekrashevich, Anton Shevtsov, Simon Karasik, Andrei Andriushchenko, Maria Trofimova, Daria Litvintseva, and Boris Yangel. SWE-rebench: An automated pipeline for task collection and decontaminated evaluation of software engineering agents. In *Advances in Neural Information Processing Systems 38*, pp. 26420–26466, 2025. doi: 10.52202/085713-0788. URL https://arxiv.org/abs/2505.20411.

Andrew M. Bean, Ryan Othniel Kearns, Angelika Romanou, Franziska Sofia Hafner, Harry Mayne, Jan Batzner, Negar Foroutan, Chris Schmitz, Karolina Korgul, Hunar Batra, Oishi Deb, Emma Beharry, Cornelius Emde, Thomas Foster, Anna Gausen, María Grandury, Simeng Han, Valentin Hofmann, Lujain Ibrahim, Hazel Kim, Hannah Rose Kirk, Fangru Lin, Gabrielle Kaili-May Liu, Lennart Luettgau, Jabez Magomere, Jonathan Rystrøm, Anna Sotnikova, Yushi Yang, Yilun Zhao, Adel Bibi, Antoine Bosselut, Ronald Clark, Arman Cohan, Jakob Foerster, Yarin Gal, Scott A. Hale, Inioluwa Deborah Raji, Christopher Summerfield, Philip H. S. Torr, Cozmin Ududec, Luc Rocher, and Adam Mahdi. Measuring what matters: Construct validity in large language model benchmarks. *arXiv preprint arXiv:2511.04703*, 2025. URL https://arxiv.org/abs/2511.04703.

Yuanhong Cai, Xiaohui Nie, Kanglin Yin, Changhua Pei, Yongqian Sun, Shenglin Zhang, Haibin Liu, Guiyang Liu, Xidao Wen, Fang Situ, and Dan Pei. A multi-dataset benchmark for evaluating LLM agents in microservice failure diagnosis. *arXiv preprint arXiv:2606.29193*, 2026. URL https://arxiv.org/abs/2606.29193.

Yinfang Chen, Manish Shetty, Gagan Somashekar, Minghua Ma, Yogesh Simmhan, Jonathan Mace, Chetan Bansal, Rujia Wang, and Saravan Rajmohan. AIOpsLab: A holistic framework to evaluate AI agents for enabling autonomous clouds. *arXiv preprint arXiv:2501.06706*, 2025. URL https://arxiv.org/abs/2501.06706.

Jackson Clark, Yiming Su, Saad Mohammad Rafid Pial, Yifang Tian, Lily Gniedziejko, Hans-Arno Jacobsen, Yinfang Chen, and Tianyin Xu. SREGym: A live benchmark for AI SRE agents with high-fidelity failure scenarios. *arXiv preprint arXiv:2605.07161*, 2026. URL https://arxiv.org/abs/2605.07161. v3; v1 2026-05-08, v2 2026-05-13, v3 2026-07-30.

Aoyang Fang, Yifan Yang, Jin'ao Shang, Qisheng Lu, Junjielung Xu, Rui Wang, Songhan Zhang, Yuzhong Zhang, Boxi Yu, and Pinjia He. OpenRCA 2.0: From outcome labels to causal process supervision. *arXiv preprint arXiv:2606.27154*, 2026. URL https://arxiv.org/abs/2606.27154.

Albert Gong, Kyuseong Choi, Abhineet Agarwal, Jason Schechner, Ryan Huang, Raj Agrawal, Anish Agarwal, and Raaz Dwivedi. ORCA-bench: How ready are language model agents for oncall? *arXiv preprint arXiv:2607.28545*, 2026. URL https://arxiv.org/abs/2607.28545.

Athira Gopal and Ashwanth Krishnan. How far can root cause analysis go on real-world telemetry data? *arXiv preprint arXiv:2607.13548*, 2026. URL https://arxiv.org/abs/2607.13548.

Saurabh Jha, Rohan Arora, Yuji Watanabe, Takumi Yanagawa, Yinfang Chen, Jackson Clark, Bhavya Bhavya, Mudit Verma, Harshit Kumar, Hirokuni Kitahara, Noah Zheutlin, Saki Takano, Divya Pathak, Felix George, Xinbo Wu, Bekir O. Turkkan, Gerard Vanloo, Michael Nidd, Ting Dai, Oishik Chatterjee, Pranjal Gupta, Suranjana Samanta, Pooja Aggarwal, Rong Lee, Pavankumar Murali, Jae-wook Ahn, Debanjana Kar, Ameet Rahane, Carlos Fonseca, Amit Paradkar, Yu Deng, Pratibha Moogi, Prateeti Mohapatra, Naoki Abe, Chandrasekhar Narayanaswami, Tianyin Xu, Lav R. Varshney, Ruchi Mahindru, Anca Sailer, Laura Shwartz, Daby Sow, Nicholas C. M. Fuller, and Ruchir Puri. ITBench: Evaluating AI agents across diverse real-world IT automation tasks. *arXiv preprint arXiv:2502.05352*, 2025. URL https://arxiv.org/abs/2502.05352.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2310.06770. arXiv:2310.06770v3; OpenReview 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. *arXiv preprint arXiv:2510.11977*, 2025. URL https://arxiv.org/abs/2510.11977.

Taeyoon Kim, Woohyeok Park, Hoyeong Yun, and Kyungyong Lee. Why do AI agents systematically fail at cloud root cause analysis? *arXiv preprint arXiv:2602.09937*, 2026. URL https://arxiv.org/abs/2602.09937.

Shanchao Liang, Spandan Garg, and Roshanak Zilouchian Moghaddam. The SWE-Bench illusion: When state-of-the-art LLMs remember instead of reason. *arXiv preprint arXiv:2506.12286*, 2025. URL https://arxiv.org/abs/2506.12286.

Qisheng Lu, Aoyang Fang, Junjielong Xu, Jin'ao Shang, Songhan Zhang, Yifan Yang, Xiaochuan Yan, and Pinjia He. Beyond fault localization: A trajectory-level study of LLM agents for microservice root cause analysis. *arXiv preprint arXiv:2608.21310*, 2026. URL https://arxiv.org/abs/2608.21310.

Evan Miller. Adding error bars to evals: A statistical approach to language model evaluations. *arXiv preprint arXiv:2411.00640*, 2024. URL https://arxiv.org/abs/2411.00640. v1, 2024-11-01.

Yichen Pan, Dehan Kong, Sida Zhou, Cheng Cui, Yifei Leng, Bing Jiang, Hangyu Liu, Yanyi Shang, Shuyan Zhou, Tongshuang Wu, and Zhengyang Wu. WebCanvas: Benchmarking web agents in online environments. *arXiv preprint arXiv:2406.12373*, 2024. URL https://arxiv.org/abs/2406.12373.

Changhua Pei, Zexin Wang, Fengrui Liu, Zeyan Li, Yang Liu, Xiao He, Rong Kang, Tieying Zhang, Jianjun Chen, Jianhui Li, Gaogang Xie, and Dan Pei. Flow-of-Action: SOP enhanced LLM-based multi-agent system for root cause analysis. In *Companion Proceedings of the ACM on Web Conference 2025*, pp. 422–431. ACM, 2025. doi: 10.1145/3701716.3715225. URL https://arxiv.org/abs/2502.08224.

Luan Pham, Hongyu Zhang, Huong Ha, Flora Salim, and Xiuzhen Zhang. RCAEval: A benchmark for root cause analysis of microservice systems with telemetry data. *arXiv preprint arXiv:2412.17015*, 2024. URL https://arxiv.org/abs/2412.17015.

Oscar Sainz, Jon Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. NLP evaluation in trouble: On the need to measure LLM data contamination for each benchmark. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 10776–10787, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.722. URL https://aclanthology.org/2023.findings-emnlp.722/.

Santiago Torres-Arias, Hammad Afzali, Trishank Karthik Kuppusamy, Reza Curtmola, and Justin Cappos. in-toto: Providing farm-to-table guarantees for bits and bytes. In *28th USENIX Security Symposium (USENIX Security 19)*, pp. 1393–1410, Santa Clara, CA, August 2019. USENIX Association. ISBN 978-1-939133-06-9. URL https://www.usenix.org/conference/usenixsecurity19/presentation/torres-arias.

Han Wang, Haoyu Li, Brian Ko, and Huan Zhang. On The Fragility of Benchmark Contamination Detection in Reasoning Models. In *International Conference on Learning Representations (ICLR)*, 2026a. URL https://arxiv.org/abs/2510.02386. arXiv:2510.02386; the arXiv comments field states "Accepted by ICLR 2026".

Yilun Wang, Guangba Yu, Haiyu Huang, Yujie Huang, Zirui Wang, Pengfei Chen, and Michael R. Lyu. Cloud-OpsBench: A reproducible benchmark for agentic root cause analysis in cloud systems. *arXiv preprint arXiv:2603.00468*, 2026b. URL https://arxiv.org/abs/2603.00468. v2; v1 2026-02-28, v2 2026-08-22.

Yuhao Wang, Zhen Qin, Xingliang Wang, Guochang Li, Weize Li, and Shuiguang Deng. EviRCA: Decoupling evidence extraction from reasoning for microservice root-cause analysis. *arXiv preprint arXiv:2609.19825*, 2026c. URL https://arxiv.org/abs/2609.19825.

Zefan Wang, Zichuan Liu, Yingying Zhang, Aoxiao Zhong, Jihong Wang, Fengbin Yin, Lunting Fan, Lingfei Wu, and Qingsong Wen. RCAgent: Cloud root cause analysis by autonomous agents with tool-augmented large language models. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 4966–4974. ACM, 2024. doi: 10.1145/3627673.3680016. URL https://arxiv.org/abs/2310.16340.

Junjielong Xu, Qinan Zhang, Zhiqing Zhong, Shilin He, Chaoyun Zhang, Qingwei Lin, Dan Pei, Pinjia He, Dongmei Zhang, and Qi Zhang. OpenRCA: Can large language models locate the root cause of software failures? In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=M4qNIzQYpd. OpenReview forum id M4qNIzQYpd.

Tianci Xue, Weijian Qi, Tianneng Shi, Chan Hee Song, Boyu Gou, Dawn Song, Huan Sun, and Yu Su. An illusion of progress? assessing the current state of web agents. *arXiv preprint arXiv:2504.01382*, 2025. URL https://arxiv.org/abs/2504.01382.

Pei Yang, Wanyi Chen, Tongyun Yang, Pengbin Feng, Jiarong Xing, Wentao Guo, Yuhang Yao, Yuhang Han, Hanchen Li, Xu Wang, Zeyu Wang, Jie Xiao, Anjie Yang, Liang Tian, Lynn Ai, Eric Yang, and Tianyu Shi. TwinRouterBench: Fast static and live dynamic evaluation for realistic agentic LLM routing. *arXiv preprint arXiv:2605.18859*, 2026. URL https://arxiv.org/abs/2605.18859.

Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. GPT-4V(ision) is a generalist web agent, if grounded. *arXiv preprint arXiv:2401.01614*, 2024. URL https://arxiv.org/abs/2401.01614.

Yuxuan Zhu, Tengjun Jin, Yada Pruksachatkun, Andy Zhang, Shu Liu, Sasha Cui, Sayash Kapoor, Shayne Longpre, Kevin Meng, Rebecca Weiss, Fazl Barez, Rahul Gupta, Jwala Dhamala, Jacob Merizian, Mario Giulianelli, Harry Coppock, Cozmin Ududec, Jasjeet Sekhon, Jacob Steinhardt, Antony Kellermann, Sarah Schwettmann, Matei Zaharia, Ion Stoica, Percy Liang, and Daniel Kang. Establishing best practices for building rigorous agentic benchmarks. *arXiv preprint arXiv:2507.02825*, 2025. URL https://arxiv.org/abs/2507.02825. v5; v1 2025-07-03, v5 2025-08-07.

A Supplementary Details

This appendix gives the full arm contract, validity-condition detail and further failure classes referenced from the body.

A.1 Full arm-contract detail

The tables below record the earlier two-arm pilot's contract and scoring discrepancies. They are not the executed P6 configuration; P6's model and budget settings are in §6, and its common-scorer requirement is in Appendix F.2.

Full arm contract — held-fixed quantities

Quantity	Value	Source artifact
Turns per run	8	`shared_profile` in the reuse manifest
Observation calls per turn	12	`shared_profile`
Diagnosis submissions	1 (one-shot)	`shared_profile`
Provider calls	8	`shared_profile`
Wall-clock limit	720 s	`shared_profile`
Output token limit	1600	`shared_profile`
Incident specification	Same sealed case, both arms	the case catalog
Fault mechanism	Same, both arms	the case catalog
Expected answer	Same private ground truth, both arms	private half of the sealed case packages
Task and topology	Condition-neutral, both arms	arm file inventories
Tool schema	Condition-neutral, side channels disabled	arm file inventories
Scoring rubric (90/100 weight)	Numerically identical rules and thresholds	per-check rubric comparison (Appendix A.2.5)

Non-identical rows

Live-arm reproduction requirement. Reproducing the live arm requires a container runtime and a host-bound artifact attestation whose **hostname, kernel, architecture and instance identifier** must match the measuring host, with no override.

Quantity	Snapshot arm	Live arm
Execution environment	None; no running system	Container subprocess + live HTTP probes
Recovery and teardown fields	Auto-true analogue (`accepted: true`)	Populated from live environment
Evidence surface dynamics	Fixed at cutoff	Live-restricted-growing
Weight-10 scorer predicate	Untimed local matcher	Time-windowed check from embedded repository
Expected labels for weight-10 check	From snapshot-side source	From live-side source

A.1.1 Provenance of the erroneous $N_{\text{EFF}} = 14$ record

The original 26-case plan recorded **14 clusters**, but its own partition lists **17**. We preserve the erroneous value as protocol history, not as a measured cluster count or the effective sample size of P6.

The partition. The pre-registration's §2.1 table (frozen pre-registration) has **17 data rows** whose size column sums to **26**. The rows partition as **5 rows of size > 1 covering 14 cases** and **12 rows of size 1 covering 12 cases**: 5 + 12 = **17** clusters, 14 + 12 = **26** cases. The document's own stated merge rule is *same application + same mechanism*; the number of distinct (application, mechanism) pairs in that table is **17** and **zero** rows share both keys, so the table is already the output of that merge and applying the rule again reduces nothing. No other rule findable in the document yields 14: application alone gives 5, mechanism alone gives 7, application × taxonomy prefix gives 11, unifying `dependency_*` per application gives 13.

Where the 14 came from. The blind case census has a §4 table of **5 rows** headed "Correlated groups:" with sizes **4 + 3 + 3 + 2 + 2 = 14**. It lists those 14 cases and omits the other 12 entirely — precisely the 12 that became the pre-registration's singleton rows. **The frozen 14 is therefore the case count of the five correlated groups**, transcribed as if it were a cluster count. The error is a category error at its origin — a count of *cases* used as a count of *clusters* — propagated by transcription, not by inference.

Historical decision and current interpretation. The earlier protocol record retained 14 to avoid a post-hoc change it regarded as favourable. That decision does not establish statistical conservatism: interval width depends on cluster membership, weights and outcomes, not the cluster count alone. No fixed square-root rescaling justifies the width of every clustered interval. The original 26-case snapshot stage was not run (Appendix F.2); 17 describes its listed partition, not an executed sample. P6 reports its own layer-specific counts in Table 2, and no interval here is computed at 17.

A.2 Validity conditions: threat-model, trust-boundary and rescoring detail

A.2.1 Condition (ii) — full threat model and trust-boundary specification

The structural argument is: pin-consulting code runs only after the entry point has executed and has already chosen which pin to consult. An in-process refusal raised as a `process-exit` exception is swallowed by a caller written to swallow it (`SystemExit` derives from `BaseException`, exit code 2 vs 0). The argument is therefore bounded to one threat model:

- **Threat:** bytes inside the tree the run reads differ, before the run starts, from what the declared pin names — whether by accident or by substitution.
- **Trusted:** the operating system, the Python interpreter, and any verifier running in a process other than the measured one.
- **Proof boundary:** a pin consulted by the entry point itself, where the entry point has already executed. Nothing else.

The words "impossibility result", "cannot be solved" and "in general" are not used. The structural statement is scoped to this threat model only. Cross-session identity of tree state — a different

question — *is* establishable and we established it: two independent implementations written from one canonical serialisation agreed byte-for-byte on every leg and on the anchor. That bound carries a limit: two derivations of one literal agree precisely when both are wrong.

A.2.2 CONDITION (II) — TREE STATE AND SCORING RECORD ARE SEPARATE FACTS

Two separately measured facts concern condition (ii) on the snapshot arm, and they must not be merged:

1. **Tree state** — three of four snapshot runs record `repository.clean: false` via an explicit flag that permits uncommitted changes, set in the snapshot run script. This concerns the *tree state* at the time the run was executed. The field counts cleanliness flags; it is not a digest comparison.

1. **Scoring record** — the snapshot arm's score file (the per-run score record) records no `producer_revision` field at all, and its run record (the run summary) has no counterpart to the live arm's digest objects (result-schema digest, the case driver's digest, tool-gateway digest, application-binary match). This concerns the *scoring record* and is a non-overlapping schema question.

The three-of-four count refers to cleanliness flags only. The missing producer revision is an entirely separate finding about the absence of a schema field. Drawing inferences across them would be a count-coincidence error of the type described in §7.2.

A.2.3 CONDITION (III) — FULL RESCORING FAILURE DETAIL

A unified scorer exists whose snapshot–live parity is certified over a named field set at a fixed scorer digest, accepted in all eight certification records. That certification covers the scorer itself, not a paired experimental result; eight certification records and eight paired-pilot runs are different populations of equal cardinality.

Rescoring both arms of both cases with the unified scorer failed for three compounding reasons:

1. The scorer refused every run on its own time-contract assertion. The two arms violate that assertion on different axes: the snapshot arm's evidence boundary does not satisfy the scorer's time-contract window, and the live arm's time contract was not recorded in a form the scorer can consume.
2. The scorer requires an evidence contract for the weight-10 check. That contract is absent for both base cases. The weight-10 check is precisely the one where the arms' predicates diverge — so the divergence needing resolution is the one the scorer cannot evaluate.
3. The unified scorer's partial score is a different quantity from the per-arm scores (different predicate for 10 weight, different expected-label source), so even a successful rescore would produce an incomparable quantity.

Verdict: making both arms scorer-comparable requires re-executing the pair under a single harness with a consistent time contract, or authoring the missing evidence contracts and re-executing. Neither is a local recomputation.

A.2.4 THE EXECUTED-VS-DECLARED RUBRIC GAP

Thirty-one predicates in the rubric declare a minimum-match threshold of five or of three (in the snapshot evaluator and in the restricted live driver). Both arms override this threshold to one at scoring time. The consequence: both arms *agree with each other* on this check while both depart from the declared rubric. This is a condition-(iii) gap under the declared-rubric leg, not a pairing defect. It is established at the first evidence level of §8 (a gap exists and is measured); the quantified consequence of honouring the declared threshold — the evidence check flipping, and the live strict-correct count moving — would be a measured effect on scores and is not established. No strict-correctness figure may be published until this gap is settled.

A.2.5 CONDITION (III) — THE COMPARABLE NINETY WEIGHT, CHECK BY CHECK

The ninety of one hundred rubric weight computed by numerically identical rules with identical thresholds on both arms is distributed as: **anomaly presence 20, causal service 30, fault category 20, root-cause time 20.** The residual weight-10 check and its two differing predicates are detailed in Appendices A.2.3 and A.2.4.

A.3 FURTHER FAILURE CLASSES, AND THE RESTRICTED-ARM SPECIMEN

A.3.1 RE-EXECUTION REVISION ALTERS THE MODEL-VISIBLE OPTION SET

Re-executing the snapshot arm at the current revision rather than at the pinned revision silently substitutes the model-visible option set. A deletion in the evidence gateway removes entries from the fault-category vocabulary it advertises; that vocabulary is injected into the enum of allowed values on the diagnosis-submission tool and hashed into the recorded tool-schema digest. Executing at a revision other than the pin therefore changes the distractor set, the difficulty and the chance level — with no trace in the produced artifacts. This maps to condition (ii) plus a pinned-revision condition the artifacts do not record. This is a reproduced check failure only: the option set changes; whether any published score changed would be a measured effect on scores and is not established. The category counts and the diff size are not reported.

A.3.2 INVOCATION NOT RECORDED

No snapshot run records its argv or the path of the producing script; the four stored snapshot runs record `environment` with only `condition` and `repository`. For the three runs whose tree was dirty (`clean: false` — pilot case A replicate 2, pilot case B replicates 1 and 2), which script produced them is not determinable: an uncommitted entry point cannot be excluded and the dirty delta was never recorded. For the single run with a clean tree (pilot case A, replicate 1), the producing script is determinable by lineage argument: the run records an earlier revision, the alternative candidate, the cross-application snapshot script, is introduced only at a later revision, and the recorded revision is not a descendant of it, so at that revision the tree contains exactly one snapshot entry point. An identity record must cover the invocation, not only the tree.

A.3.3 EXECUTED THRESHOLD DEPARTS FROM THE DECLARED RUBRIC

See Appendix A.2.4 above. The executed-vs-declared rubric gap under condition (iii) is the same fact; it surfaces in §7 as a named failure class because it is one of the three questions our own artifacts cannot fully answer.

A.3.4 THE RESTRICTED ARM: DECLARED BY A LABEL, CHECKED BY A TAUTOLOGY, RECORDED NOWHERE

Measured at a single fixed revision of the pipeline.

What asserts the restriction. A fixed-mode and a growing-mode run report differ, in the window domain, by exactly one key: `acceptance.agent_isolation.cross_app_observation_mode`, written straight from the CLI `-observation-mode` argument by the live driver. It records declared intent and is never a measurement of the gateway.

What checks it. The grounding record's `initial_cutoff` is arm-invariant: the live driver computes `injection + W` unconditionally, with no reference to `mode` — unlike another branch of the same driver, which correctly emits `""` for non-fixed — so a growing-mode run writes the same concrete timestamp as a fixed-mode run. The scorer check that consumes it compares one recomputation of `injection + W` in the semantic scorer against another computed the same way, with `FAULT_WINDOW_SECONDS = 29` in the scorer equal to `W_SECONDS = 29` in the live driver: it cannot fail, and `initial_cutoff == expected_initial_cutoff` in 100% of emitted certification records. The one binding check exempts the live arm — the semantic scorer ties `window_end` to the cutoff only for `TRACK_SNAPSHOT`.

What gates it, and what tests it. The offline preparer's `L-fixed` gate is a source-substring check, satisfied merely by the gateway line that reads the cutoff existing; it cannot observe a runtime-empty value. The gateway's unit test for fixed-window evidence growth passes the cutoff as a literal argument, bypassing the env wiring that can fail, and the test file's shared fixture omits `observationCutoff` entirely, so every integration-style test in that file passes with the restriction disabled. Our own pre-registered auditability screen adds little: the harness fires six authenticated requests before the agent runs, so a non-zero-byte tool audit is satisfied by our own smoke traffic even with zero agent queries and an empty cutoff.

What is recorded. At the measured revision, nothing that identifies the cutoff the gateway held. The value is read with a bare `os.Getenv` — the only field in that struct literal not read through the gateway's required-value or default-value environment helpers — and is read back only by the gateway's window-bounding function. Every cutoff-shaped value in an emitted artifact is an independent recomputation of `injection + W` by the orchestrator or the capture tool. The audit does carry a resolved window, but the resolved `end` equals the cutoff only when the agent omitted `end` and the cutoff was non-empty; with an empty cutoff the gateway substitutes `time.Now()`, which is artifact-indistinguishable from an enforced window.

Two qualifications, stated so that the specimen is not overstated. An *unparseable* cutoff is fail-closed, not silent: the gateway returns `invalid observation cutoff` and every query is rejected; only the *empty* case is silent. And the skip is partial: the cutoff check is bypassed, but a later `endTime.After(now)` check still bounds the window at wall-clock + 5 s, so a bound formally remains — what is absent is the restriction.

Remedy. Emit the configured value on every audited line — the `record` map in the gateway's audit function, where the gateway receiver is already in scope — so that the audit covers rejections and empty result sets, which the evidence-payload path does not. This is instrumentation, not repair: nothing about the threshold, the default or the fail-open behaviour is changed by it.

Scope. A reproduced check failure and no further: an artifact whose identity is unclear passes every check that exists. **A measured effect on inputs, samples or scores is conditional and not asserted here. A measured effect on a ranking or the Snapshot–Live conclusion is not claimed: the three-arm experiment (§6) does not isolate this defect.** This is a within-framework specimen; no detection rate is claimed, and the in-repo counts above carry no cross-framework generality.

A.4 Numbers not to be read as effect sizes or detection rates

Each number below is measured; the table states the reading it does not support.

B Additional Protocol Details

This appendix gives further related work (Section 2) and protocol details referenced from Section 4.

B.1 Further related work

Benchmark validity. Flaws in task setup, reward design and metrics are known to undermine what agent and LLM benchmarks measure (Zhu et al., 2025; Bean et al., 2025), and harness-level defects in agent evaluation are a recognised problem (Kapoor et al., 2025). Our validity conditions address this concern for one comparison: a paired Snapshot–Live difference.

RCA agents and benchmarks. The systems such benchmarks evaluate include tool-augmented autonomous agents for cloud RCA (Wang et al., 2024) and SOP-enhanced multi-agent RCA (Pei et al., 2025). RCAEval provides datasets and an evaluation environment for microservice RCA (Pham et al., 2024); successors of the snapshot paradigm add step-wise causal ground truth (Fang et al., 2026) and expert-labelled evaluation of the reasoning process (Cai et al., 2026). Process-level analyses find that final-answer checks do not show why the reasoning failed (Kim et al., 2026) and that answer correctness and diagnostic quality diverge (Lu et al., 2026); concurrent work that decouples evidence extraction from reasoning places the bottleneck in judging evidence already surfaced rather than in searching for it (Wang et al., 2026c). Operations-agent benchmarks also span IT automation tasks beyond RCA, including SRE scenarios (Jha et al., 2025).

Number	Where it is real	Why it must never be quoted
`22/29` (mutation suite)	our mutation suite (Appendix C.5)	**Never a detection rate.** It is our own suite scored against a test set constructed around problems we had already found. The paper cites only the silent-four fact, and the mandatory qualifier travels with any use.
"26 nominal base cases collapse to 14 application x mechanism clusters" (blind case census, §4 prose)	The census sentence exists and is quoted verbatim in the pre-registration record	**Never quotable as a measurement.** The census's 14 is the case count of its five correlated-group rows (4+3+3+2+2), and the census never enumerates 14 clusters or lists a singleton cluster. The partition under the stated merge rule is 17 (Appendix A.1.1). The sentence may be quoted only as the *provenance* of a transcription error, never as a clustering result.
One catalog case's `latency_ratio = 311.0`	its validation record, `checks.resource_case_analysis.propagation_bundle.service_request_impact`	**Never an effect size.** The baseline p95 is `0.0` ms, so the ratio is computed against the substituted denominator floor `LATENCY_DENOMINATOR_FLOOR_MS = 1.0`, not against a measured baseline; pick a different floor and the ratio moves arbitrarily. Acceptance does not depend on it — that record's fault p95 of **311 ms** clears the `minimum_fault_p95_latency_ms = 20.0` floor independently. "311× amplification" is an artifact of the floor substitution. (The sibling `downstream_request_impact` record on the same case carries `latency_ratio = 312.0` against a fault p95 of 312 ms, under the same floor and the same prohibition.)

Offline and online evaluation in other agent domains. For web agents, offline evaluation on cached websites and online evaluation on live websites coexist (Zheng et al., 2024), online evaluation frameworks respond to benchmarks that capture only static aspects of the web (Pan et al., 2024), and a rigorous online assessment suggests over-optimism in previously reported results (Xue et al., 2025). An agent-routing benchmark pairs a static track for fast offline iteration with a live-execution track for end-to-end validation (Yang et al., 2026).

Integrity of frozen benchmarks. Data contamination must be measured for each benchmark (Sainz et al., 2023). For the frozen coding tasks of SWE-bench (Jimenez et al., 2024), gains on its verified subset may be partly driven by memorization (Liang et al., 2025), static tasks go stale through contamination (Badertdinov et al., 2025), and passing tests can be too weak to verify a patch (Aleithan et al., 2024). Each is a pass that does not certify the property it reports, the failure shape of Section 2.

B.2 Scope of the budget profile

Both pilot cases ran under one frozen budget profile (Appendix A.1)—a property of those pilot pairs, not of the benchmark at large. P6's settings are reported separately in §6.

B.3 Clustering level and the interval floor

The original 26-case plan's recorded count of 14 clusters is a transcription error; its listed application–mechanism partition has 17 (Appendix A.1.1). Neither is the executed P6 sample size. P6 uses the layer-specific cluster counts in Table 2. Treating correlated cases as independent can understate uncertainty; no fixed interval-width ratio is implied. Our reporting rule withholds intervals below eight clusters.

C Supplementary Details on Validity, Pilot Runs and Failure Classes

This appendix holds detail referenced from Sections 5 and 7 and Appendix D.

C.1 Condition (ii) on the paired pilot data

This continues the paired-data paragraph of §5.2. For the single run with a clean tree (pilot case A, replicate 1), the producing script is determinable — but by a lineage argument, not by recorded provenance: the alternative candidate script was not introduced until after the recorded revision, so exactly one snapshot entry point existed at that revision. For the three dirty-tree runs, the producing script is not determinable at all; recoverability is a property of the tree state, not of the artifact. An optional schema field makes an identity leg unfailable: twenty-one declared, one mismatching and one undeclared across twenty-three oracle configurations. The invocation record behind the lineage argument is in Appendix A.3.2.

C.2 Condition (iii): a lineage reconstruction for one pilot case

Condition (iii) held, and no artifact can show it. For pilot case A the Live arm records an evaluator revision and the Snapshot arm a repository revision; the former is an ancestor of the latter, and the one commit between them changes 38 files, all additions, none of them Python, Go or shell files: all but one are report files, the remaining one a documentation file. The scorer code was identical across the arms. But the lineage was reconstructed from two revisions recorded for other purposes, one of which the record never offers as a scorer identity: a finding, never a certification — the pre-declared check still cannot be run and the count passing (iii) remains zero. A complete, well-formed, gate-passing record cannot distinguish a condition that held from one that did not; here the silent condition was satisfied.

C.3 The admissibility ledger

An admissibility ledger over the twenty-six eligible base cases. The answer-uniqueness half of the admissibility screen passes twenty-six of twenty-six; its reachability half is undeterminable on every arm, the repository holding no reachability predicate and no artifact that records one. The evidence-sufficiency gate reads a semantic-evidence contract unconditionally, and that file is absent from thirteen of the twenty-six sealed case packages, so for those it raises before any gate logic runs. Three sealed case packages — the whole `memory_pressure` cluster — roll a false `error_rate_accepted` sub-check up into a true acceptance block, and an independent batch gate caught on one of them what that acceptance hid. Only the third is a reproduced check failure — an artifact whose failing leaf passes its own certification; the first two void condition (i) by being unanswerable rather than by failing. Their effect on model-visible inputs, the valid-sample set or scores is not measured here. The three findings are the last three condition-(i) rows of Table 3.

C.4 Results the pilot record cannot give

Table 4 lists the results the protocol specifies, what blocks each on the pilot record of Appendix D, and where the three-arm experiment reports it.

C.5 Instrument defects in our own pipeline

Twelve instrument defects arose in one round; text-keyed instruments failed, structure-keyed ones held, each caught by a second instrument or downstream contradiction; for four of the seven defects our mutation suite missed, it was completely silent. Two conclusions were retracted after measurement, one where the auditor had inspected the modules a refusal named, never the entry point that ran. On the specified `L-fixed` arm, at the measured revision, the restriction is recorded only as a CLI label whose check compares one recomputation with another; no artifact records the gateway's cutoff (Appendix A.3.4).

Of the twelve registry-wired case drivers, one is enumerated and start-time enforced, eleven pass the supporting-module check, none refused — twelve counts case drivers, not instrument defects. Their effect on inputs, samples, scores or rankings is not measured here.

Table 4: Pre-specified results: what blocks each on the pilot record, and where the three-arm experiment (§6) reports it.

Result	On the pilot record; in the three-arm experiment
Headline Snapshot–Live agreement over N base cases	One-revision scoring (below); N = 1; clustering floor. Reported per layer (Table 2)
What restricting the live surface changes	No `L-fixed` run in the pilot. `L − L-fixed` per layer (Table 2)
The attributable comparison, `S − L-fixed`	As the previous row, plus one-revision scoring. Reported per layer (Table 2)
Both arms scored at one scorer revision (one-revision scoring)	Snapshot arm records no producer revision; needs re-execution, not recomputation
A repaired paired N	Snapshot arm must be re-executed from a clean checkout
Residual difference with the agent held fixed	Requires one-revision scoring; a deterministic oracle is not an agent
Measured effect on inputs, samples or scores	Not established
Measured effect on rankings or the Snapshot–Live conclusion	Not established; the three-arm experiment does not isolate a failed condition (§8)

D PILOT MEASUREMENTS AND ADMISSIBILITY

This appendix holds the descriptive pilot that preceded the three-arm experiment of Section 6. Section 5 leaves one base case passing conditions (i) and (ii), none passing all three, and condition (iii)'s stop rule leaves that pair uncomparable across arms.

One base case passes conditions (i) and (ii) today; under a stricter rule requiring both replicates identity-clean, none does. None passes all three: (iii) cannot be evaluated for any pair. Condition (ii) is the cause: three of four snapshot runs record a dirty worktree, so the declared revision does not identify the executed bytes, and the delta was never recorded. N = 1 is a property of the stored record, not a sampling choice.

Forty-one of fifty-three sealed case packages carry a complete snapshot-scorable input and all fifty-three a live-runnable specification, yet without new collection zero additional pairs are constructible; the twenty frozen recordings have an oracle on the live side that read the private ground truth, so they yield none. The audited ceiling is two. The binding resource is identity-clean live-capable base cases, not compute. The admissibility ledger over the twenty-six eligible base cases is in Appendix C.3.

The eight pilot runs, described rather than combined. Condition (i) passes on both arms of both cases. The live arm recorded four successes at full marks and the snapshot arm two, the two snapshot failures scoring eighty and fifty. Table 5's validity column keeps the three outcomes apart: none undiagnosable, three runs incomparable, one genuine error. The one identity-clean snapshot run is one of those two failures — reported rather than smoothed.

The arms are reported side by side and not subtracted, as the stop rule of §5.3 requires; no cross-arm quantity is stated for these pilot pairs. The three-arm experiment's paired score differences, and the agreement and flips over usable replicate pairs are in Section 6.

Where the arms diverge, at the only resolution available. Divergence in these pairs is confined to the categorical attribution checks — fault category on one case, causal service and fault category on the other — while anomaly presence, root-cause time and evidence agree: an observation over two cases carrying one mechanism each, latency and configuration, never a per-mechanism rate. Its pre-registered falsifier travels with it: uniform divergence across all five checks signals a scorer mismatch, not an environment effect. The reporting rule — intervals clustered at the application $\times$ mechanism level, sign convention `S − L-fixed` — is not exercised: our pre-registration forbids clustered intervals below eight clusters, and at N = 1 they are undefined.

This appendix measures no effect on inputs, samples, scores or rankings (Appendix C.4).

Table 5: The eight pilot runs, with the validity column that governs them.

Runs	Arm and tree state	Cond. (ii)	Reportable as	Measured outcome
4	Live	PASS	Per-run description only	Four successes at full marks
1	Snapshot, clean worktree (pilot case A, replicate 1)	PASS	The only identity-clean snapshot cell; producing script determinable by lineage, not by record	One of the two snapshot failures
3	Snapshot, dirty worktree admitted by flag	FAIL, silently	**INCOMPARABLE** — score well-formed, executed bytes unidentified	Not reported as outcomes
8	Both arms, both cases	—	Condition (i): **PASS**	Full marks reachable from the snapshot surface alone
—	Both arms	Cond. (iii)	**No cross-arm number**	Stop rule: snapshot producer revision absent

E SUPPLEMENTARY P6 RESULT TABLES

The joint analysis has pre-declared layers (Appendix F.3): J1 as pre-registered (r15); J2 and J3, amended and extended, over all admitted units; J2s and J3s, their sensitivity layers; and J5, tier U with the primary units, also split by the P4 pattern of sufficient arms and never pooled into J1–J3. Tables 6 and 7 are referenced from Section 6. Table 6 breaks the primary layers J1 and J3, and J5, out by pre-registered model. In Table 7, each row is one wave, exploratory model and arm; rows are never pooled across waves, with each other or with the pre-registered models (Appendix F.4, "Waves").

Table 6: Per pre-registered model: S − L-fixed point, aggregated strict-outcome agreement computed from usable replicate pairs, and strict-outcome flip counts over those pairs (S→F: success to failure, F→S: failure to success). Agreement and flips are descriptive diagnostics, not the case-level estimand of Definition 3 (§3.2); replicate pairs add no independent cases. The recorded H1–H2 verdicts concern S − L-fixed; H3 concerns L − L-fixed. J5 includes tier U and is not attributable to evidence timing. Brackets give intervals where the reporting floor is met; a printed zero bound may be positive before rounding. H1 verdicts from the score-interval clause are protocol history, not evidence about majority strict-outcome agreement. Under the recorded H2 rule, an interval covering zero is labelled falsified; statistically it only leaves the direction unestablished.

Layer	Model	S − L-fixed	Agree	S→F	F→S	H1	H2	H3
J1	1	+8.3 pp	100.0%	0	0	`not_evaluated`	`not_evaluated`	`not_evaluated`
J1	2	+0.0 pp	100.0%	0	0	`not_evaluated`	`not_evaluated`	`not_evaluated`
J1	3	+8.3 pp	75.0%	0	1	`not_evaluated`	`not_evaluated`	`not_evaluated`
J3	1	+20.8 pp	75.0%	2	0	`not_evaluated`	`not_evaluated`	`not_evaluated`
J3	2	+0.0 pp	100.0%	0	0	`not_evaluated`	`not_evaluated`	`not_evaluated`
J3	3	+4.2 pp	87.5%	0	1	`not_evaluated`	`not_evaluated`	`not_evaluated`
J5	1	+9.4 pp [−9.4, +27.1] pp	78.1%	5	1	`not_falsified`	`falsified`	`not_falsified`
J5	2	−4.8 pp *n/a (below floor)*	100.0%	0	0	`not_evaluated`	`not_evaluated`	`not_evaluated`
J5	3	+27.1 pp [+0.0, +51.0] pp	62.5%	4	4	`falsified`	`not_falsified`	`not_falsified`

J5 verdicts. The J5 H1, H2, H3 verdicts in Table 6 are, for model 1, `not_falsified`, `falsified` (under the H2 rule, because its interval covers zero: the direction is not established), `not_falsified`; for model 3, `falsified` (by the recorded H1 interval clause: its lower bound is above zero but rounds to zero as printed; this is protocol history, not evidence about majority strict-outcome agreement), `not_falsified`, `not_falsified`; and for model 2, below the floor on J5 (7 clusters against 8), `not_evaluated`, `not_evaluated`, `not_evaluated`; they describe the pooled tier and are not attributable to evidence timing.

Exploratory models. Table 7 describes the exploratory models per wave, arm and model: valid entries, mean score and strict-correct rate. They are never pooled across waves, with each other or with the pre-registered models, and carry no contrast, interval or verdict. Descriptively and per wave, exploratory model 3 scores higher under S than under L-fixed and L in each of x2, r15u and x2u, its live-arm means at the bottom of the scale in all three (the cause of this live-arm floor was not investigated and is not attributed to evidence timing); model 1's S mean equals its L-fixed mean in x2 (L-fixed entries partly missing), is higher in r15u and lower in x2u; r18u ran S only.

Table 7: Exploratory models, descriptive only: per wave, never pooled across waves or into any layer.

Wave	Model	Arm	Valid/entries	Mean score (%)	Strict-correct (%)
x2	EXPLORATORY_SLOT_1	L	4/4	91.7%	75.0%
x2	EXPLORATORY_SLOT_1	L-fixed	2/4	100.0%	100.0%
x2	EXPLORATORY_SLOT_1	S	4/4	100.0%	100.0%
x2	EXPLORATORY_SLOT_3	L	4/4	0.0%	0.0%
x2	EXPLORATORY_SLOT_3	L-fixed	2/4	0.0%	0.0%
x2	EXPLORATORY_SLOT_3	S	4/4	58.3%	50.0%
r15u	EXPLORATORY_SLOT_1	L	6/6	33.3%	0.0%
r15u	EXPLORATORY_SLOT_1	L-fixed	6/6	38.9%	0.0%
r15u	EXPLORATORY_SLOT_1	S	6/6	44.4%	33.3%
r15u	EXPLORATORY_SLOT_3	L	6/6	0.0%	0.0%
r15u	EXPLORATORY_SLOT_3	L-fixed	6/6	0.0%	0.0%
r15u	EXPLORATORY_SLOT_3	S	6/6	83.3%	50.0%
r18u	EXPLORATORY_SLOT_1	S	6/6	88.9%	83.3%
r18u	EXPLORATORY_SLOT_3	S	6/6	100.0%	100.0%
x2u	EXPLORATORY_SLOT_1	L	2/2	50.0%	50.0%
x2u	EXPLORATORY_SLOT_1	L-fixed	2/2	100.0%	100.0%
x2u	EXPLORATORY_SLOT_1	S	2/2	50.0%	50.0%
x2u	EXPLORATORY_SLOT_3	L	2/2	0.0%	0.0%
x2u	EXPLORATORY_SLOT_3	L-fixed	2/2	0.0%	0.0%
x2u	EXPLORATORY_SLOT_3	S	2/2	50.0%	50.0%

F DEVIATIONS FROM THE PRE-REGISTRATION

This appendix records departures from the pre-registration in P6. The pre-registration was amended eight times, on 2026-09-23 and 2026-09-25, and six controller rulings changed how its rules were applied. For each departure we state what changed, when it changed relative to the model runs, why, and what it costs the inference. All times are UTC. The two remote execution hosts are called host 1 and host 2. Operational counts come from execution sentinels and the control ledger: runs planned, completed, failed, skipped, units admitted, hosts and spend. Outcome estimates and hypothesis verdicts are reported in Section 6 and Appendix E; this appendix also records the analysed cluster counts and layer statuses.

F.1 TERMS USED IN THIS APPENDIX

- A *unit* is one case, captured once without any model in all three arms (S, L-fixed, L). The capture produces three cells.
- The admissibility review (P4) has two blind reviewers and an adjudicator. It rates each cell sufficient or insufficient. A unit is *admissible* if all three of its cells are sufficient.
- A *wave* is one capture generation with its own execution revision and its own P6 run.
  - r15 is the nine Formal-9 units at the first accepted execution revision.
  - r18 is the re-capture of Amendment 5.
  - x1 and x2 are extension Waves A and B of Amendments 6 and 7.
  - The suffix *u* names a wave's tier-U units (Amendment 8).

- The *origin run* of a wave runs all three arms of a unit through one driver on the unit's capture host.
- The *S section* runs the S arm of record separately, on the local machine, without contacting a host (ruling DEV-S1, below).
- P denotes the three pre-registered models. X denotes the exploratory models.

F.2 Amendments 1 and 2 (signed before any model run)

Amendment 1 was signed on 2026-09-23 at 14:17Z by the study owner. Amendment 2 was signed on 2026-09-23 at about 14:53Z, also by the study owner, and entered in the ledger at 14:55Z. Both were signed before any model run and before any of the nine units' no-model captures (twenty-seven three-arm cells) had been made. Neither amendment changes the hard gates listed under run order and truncation.

The snapshot stage over all eligible cases is not run. The pre-registration scheduled three stages on different case sets:

- first, the Snapshot arm on all twenty-six eligible base cases;
- then, Live-fixed on a pre-selected subset of nine cases in nine distinct clusters, paired with their first-stage Snapshot runs;
- finally, the full Live arm on the same nine.

Its stopping rule 3 froze the snapshot stage, and did not start the Live-fixed stage, if the snapshot stage was not complete forty-eight hours before the full-paper deadline. The first amendment was signed before that rule would have triggered. The snapshot stage over twenty-six cases is not run. Instead, all three arms (S, L-fixed, L) run only on nine units. Each unit's own S run is the pairing partner for its L-fixed and L runs. Consequently this paper contains no snapshot-side capability estimate over the twenty-six cases. It also contains no rescoring at that scope that could measure an effect on model-visible inputs, the valid-sample set or scores.

Admissibility of the nine units. Pre-registered stopping rule 5 withholds the Live-fixed stage when the pre-run ledger shows fewer than eight clusters constructible with both arms. That ledger, over the twenty-six cases, found zero. So the original Live-fixed plan was not started, and it remains unstarted. Stopping rule 5 constrains only that plan.

The nine units are rebuilt cases in which all three arms are constructible by design. Two of them belong to the pre-selected subset and the other seven to its pre-registered reserves. They were not chosen by the reserve rule (next paragraph). They fall in eight independent clusters across four applications, exactly at the floor of eight (zero margin). Losing any one cluster, to inadmissibility or truncation, therefore removes interval estimation under the pre-registered rule.

Their admissibility is decided by a review of cells captured without any model, and that review is frozen before any model run:

- If the review is negative overall, the main experiment is not run.
- Otherwise, models run only on the admissible units.
- Each inadmissible unit is listed with its reason before any model run and counted as pre-registered attrition.
- If the admissible units cover fewer than eight independent clusters, the runs still proceed, but only descriptive counts are reported and no intervals.

Selection of the nine units. The nine units were selected after the pre-registration was frozen, by a separate eligibility census. The census rule was frozen before it was applied, and it was applied without reading any model score. Under the literal reading of one of its criteria, the rule admits no case. The reading the agents adopted, also after the freeze, admits these nine.

The units were not selected by the pre-registered reserve rule. That rule substitutes a reserve only within the same mechanism, in the stated order, and only for a case that fails a screen. The change of selection rule has four effects:

- It replaces seven of the nine pre-selected cases.
- It changes the mechanism balance from three cases per mechanism to five dependency-unavailability cases and two each of wrong-endpoint and dependency-latency cases. Within-mechanism substitution cannot produce that balance.
- It drops one application and adds another, so the units still span four.
- It reduces nine distinct clusters to eight, because two units share one cluster.

Arm realizations and per-run validity. The three arms do not share one realization:

- S uses the case's sealed capture.
- L-fixed and L are each run on a fresh live realization of the same case.
- L's telemetry grows during the investigation.
- L-fixed is clamped at the gateway's own cutoff.

The protocol requires the gateway to record that cutoff in the run's audit. It does not do so at the revision measured in Appendix A.3.4, and an L-fixed run without that record is undeterminable.

The admissibility review certifies only the identity and contract of each unit. The validity of each L-fixed and L run is judged run by run, by the three conditions of Section 5:

- a run whose realization fails condition (i) is undiagnosable;
- a run that fails (ii) or (iii) is incomparable;
- neither is counted as a model failure.

Run order and truncation. Time control for the nine units moves from the stopping-rule-3 freeze to a pre-declared run order and truncation rule. The main analysis keeps the pre-registered three models, with two replicates per arm and case. The first amendment requires that, in each case-and-model block, S and L-fixed run before L.

Before any run, the agents fixed the order as follows:

- A block is one case, model and replicate.
- The order is replicate-major: every first-replicate block runs before any second-replicate block.
- Within a block, S, L-fixed and L run in that order.
- The case order and the model order are fixed before the first run.

Blocks not started by the data freeze are dropped whole, in the declared order. Only complete pairs are analysed, and the number dropped is reported. The truncation time was first 2026-09-26T02:00Z. Amendments 5 and 8 later moved it (below).

Stopping rules 1, 2, 4 and 6 continue to apply. The budget thresholds of rules 1 and 2 change as described below. The hard gates are unchanged:

- the case set is fixed at nine;
- every arm is scored by the same scorer at the same revision;
- every run declares its evaluation mode and the end of its observation window explicitly;
- no direct fault label is exposed;
- all three arms are the same case;
- checkout verification cannot be skipped;
- the snapshot model runner refuses, fail-closed, any case whose sealed evidence is not declared requalified under the executing revision;
- a cost ledger is kept.

The first hard gate was later relaxed by Amendments 5–8 (below). Results involving the L arm, and the secondary hypotheses, remain exploratory, as pre-registered.

Build isolation without a separate daemon. No separate, experiment-owned daemon is used. The gateway image is built in every case lifecycle from the pinned source, on the host's default builder. Objects the experiment creates are recorded by their exact identifiers. Clean-up removes only those identifiers, and no prune without a filter is run. Before and after its work, each host records inventories of images, build cache, containers, networks and volumes. These show that objects the experiment did not create are unchanged.

Technical reruns. The canary is a pipeline check run before the main experiment, and it is never sealed as evaluation data. After a technical failure it may be rerun, without any score being inspected. A cell of the main experiment may be rerun only when a technical-rerun eligibility policy, frozen in advance, classifies it as eligible:

- the policy fails closed;
- it allows at most two reruns per cell;
- each rerun is entered in the ledger before it executes;
- no cell is rerun because of its outcome.

In execution, the policy's rerun budget was zero. No P6 run was rerun.

Cost ceiling. The cumulative ceiling on model and infrastructure cost (stopping rule 1) was raised tenfold, to CNY 10,000. The per-case-and-agent warning and hard-stop thresholds of stopping rule 2 were scaled by the same factor, to CNY 250 and 500. Stopping rule 4 is unchanged.

If the first five runs use more than twice the assumed tokens, the cost projection is recomputed mechanically from measured tokens and entered in the ledger. The runs continue while the projected cumulative cost stays within the raised ceiling. The cost ledger is mechanical: token counts returned by the provider multiplied by a price table, with no manual price comparison. Up to four heavy workers may run in parallel.

Additional exploratory models. The canary uses the three models of the frozen specification. Any further model is exploratory. Before any run, the agents froze three such models and their order, and entered them in the run matrix before the first run of the main experiment. Each runs only after every block of the three pre-registered models, and only after passing a technical sanity check on a development case. Their results are reported separately and descriptively, and are never pooled with those of the pre-registered models.

Corrections to the pre-registration text. Before these amendments, the frozen pre-registration received versioned corrections. Each was issued as a successor document rather than an edit, and all came before any model run of the three-arm experiment. The provenance of $N_{\text{eff}} = 14$ (Appendix A.1.1) is one of the questions they address.

F.3 Amendments 3 to 8

Amendments 3 to 8 were signed by the controlling agent, under a standing delegation from the study owner given on 2026-09-23 at about 14:52Z. The owner's directives prompted Amendments 5 to 8: the case count was too small to support the paper. Table 8 summarizes all eight amendments. None of them was written after any P6 score, diagnosis or grounding verdict had been read.

Amendment 3: the P4 decision rule.

- *What.* Each cell is rated by two blind reviewers on a binary scale (sufficient or insufficient). A disagreement, or any not-reviewable rating, goes to a third adjudicator. A unit is admissible if and only if its S, L-fixed and L cells are all finally sufficient. The P4 verdict is Go if at least one unit is admissible. Fewer than eight clusters covered means descriptive results only. Inter-rater agreement is reported but is not a gate. An inadmissible unit is not re-captured or re-reviewed.
- *When.* Signed on 2026-09-23 shortly before 22:26Z, when its wording was revised. At that time the cells were still being captured, no cell had been reviewed, and no model had run.

Table 8: The eight amendments. "Runs in the affected wave" says whether any model run of the units the amendment governs existed when it was signed.

No.	Signed (UTC)	Rule it replaced	Runs in the affected wave
1	09-23 14:17	Stopping rule 3 and the 26-case snapshot stage; three arms on nine units instead	None anywhere
2	09-23 ≈14:53	Daemon isolation; rerun rule; cost ceiling (rules 1–2, tenfold); exploratory models	None anywhere
3	09-23 <22:26	Undefined P4 Go/No-go and admissibility (left open by A1)	None; 0 of 27 cells reviewed
4	09-23 22:55	One capture attempt per unit (one host reassignment added); P4 batch split	None; after capture, before P4
5	09-25 04:48:53	A3's no-re-capture rule for inadmissible units; truncation 02:00Z → 03:00Z	r18: none. r15 runs done, no score read
6	09-25 04:48:53	Fixed case set of nine; extension pool (Waves A, B)	x1, x2: none; nothing captured
7	09-25 05:06:54	A6's pool (aligned to the pre-registered 26-case list)	x2: none; nothing captured
8	09-25 13:45:30	Admissibility as a gate for all runs (tier U added); review gating; truncation 03:00Z → 05:00Z	Tier U: none. r15 results sealed, unopened

- *Why.* Amendment 1 left "P4 Go/No-go" and "admissible" undefined.
- *Cost.* None beyond Amendment 1. The rule was fixed before any cell was reviewed.

Amendment 4: pre-capture failures and host reassignment.

- *What.*
  - A4-1: a unit whose capture failed before any cell was made, because of a host-local artifact, may be captured once more on the other host. This was applied only to case-0602, which was captured on host 1.
  - A4-2: case-0509 is not repaired. It has no cells, so it is inadmissible, and its cluster is lost.
  - A4-3: the P4 review runs in batches.
- *When.* Signed on 2026-09-23 at 22:55Z, after capture had reported on both hosts and before any P4 reviewer was dispatched. No model had run.
- *Why.* Two of the nine lifecycles failed before capture, for reasons that were not properties of the case.
- *Cost.* Losing case-0509's cluster leaves at most seven of eight clusters. From this point, the nine-unit design could no longer reach the interval floor.

Amendment 5: one re-capture of the seven inadmissible units. The P4 review of r15 (adjudicated 2026-09-23T23:27:13Z) admitted two of nine units, case-0602 and case-0605. They cover two clusters in one application.

- *What.* The seven inadmissible units may be re-captured once, with instrumentation-only changes, at a new execution revision. Those seven are case-0502, 0507, 0508, 0509, 0614, 0615 and 0616.
  - Only case-0502, 0507 and 0508 were re-captured (wave r18).
  - The three hotel units, case-0614, 0615 and 0616, were bounded by a three-hour timebox. They kept their first cells.
  - Case-0509 was not re-captured.

  The amendment adds three analysis layers:
  - as pre-registered (case-0602 and 0605 only);
  - amended (all admissible units);

  - a sensitivity layer restricted to the new revision.

  Stop rule 6 is evaluated separately for the new units. Truncation moves to 2026-09-26T03:00Z.
- *When.* Sealed on 2026-09-25 at 04:48:53Z. The r15 origin runs and S sections were complete by then. No P6 screen had been computed and no score had been read. No model had run on the seven units.
- *Why.* Two units in two clusters cannot support the paper's comparison. The owner directed a re-capture if it was faster than finding new cases.
- *Cost.* The amendment breaks Amendment 3's no-re-capture rule. It selects the units to re-capture on a pre-model P4 verdict. Units from two execution revisions become a between-unit factor, which the sensitivity layer removes. In the event, the r18 P4 review admitted none of the three re-captured units (2026-09-25T15:01:50Z, NoGo, 0 of 3). So the amended and sensitivity layers add no primary unit.

Amendment 6: extension units.

- *What.* New units come from the frozen cross-application case list, minus the Formal-9 cases: fourteen case ids. Wave A has ten and Wave B has four.
  - Selection is by feasibility only (code facts, no subject outcome). It is recorded before capture.
  - A case not ready at its timebox is deferred by the clock alone.
  - Each unit gets one capture attempt, at a new execution revision per wave.

  The amendment adds an extended analysis layer over all admissible units, and a sensitivity layer.
- *When.* Sealed with Amendment 5, on 2026-09-25 at 04:48:53Z. No extension unit had been captured, and no model had run on one.
- *Why.* The same shortage of admissible clusters.
- *Cost.* The case set is no longer the pre-registered nine. The extended layer mixes applications and revisions. Wave A yielded one captured unit, case-0604 (wave x1). Its P4 review found it inadmissible on 2026-09-25 at 12:45:30Z (NoGo, 0 of 1).

Amendment 7: pool correction.

- *What.* The pool is aligned to the pre-registered twenty-six-case list. Amendment 6 had dropped case-0501, 0611 and 0613 by a controller error. They are added to Wave B (wave x2).
- *When.* Sealed on 2026-09-25 at 05:06:54Z. It was written after the Wave A feasibility table, which holds only code facts. No extension unit had been captured, and no model had run on one.
- *Why.* To correct the error in Amendment 6.
- *Cost.* None beyond Amendment 6. The three cases are on the pre-registered list. The x2 P4 review admitted case-0501 and case-0613 (2026-09-25T19:33:29Z, Go, 2 of 3, two clusters, one application).

Amendment 8: relaxations to obtain more cases.

- *What.*
  - A8-1 creates tier U. A unit is in tier U if it is in the Amendment 5–7 pool, its one capture attempt produced all three cells, and it is not admissible. For tier U, the P4 verdict is only a stratification label (the pattern of sufficient arms). Tier-U units run the unchanged P6 design and form analysis layer 5 (J5). Layer 5 is never pooled into layers 1–4.
  - A8-2: a wave's review no longer blocks its capture or P4. Its P6 still needs an accepting review.

  - A8-3 sets a 45-minute review timebox, with a hard stop at 60 minutes, and allows a mitigated acceptance of a major finding. The finding must not touch the scorer, ground truth, contract, fault, evidence, projection or model view.
  - A8-4 lets waves run in parallel on the two hosts.
  - A8-5 moves truncation from 03:00Z to 2026-09-26T05:00Z.
  - A8-6 allows a semantic contract to be added to a Wave A unit, but only from a contract already committed.

  Four rules are not relaxed:

  - one capture attempt per unit;
  - no selection, exclusion or rerun on any outcome;
  - the same scorer, gateway, adapter and projection bytes for every unit;
  - blindness.

- *When.* Sealed on 2026-09-25 at 13:45:30Z. The r15 outputs existed and were sealed, but unopened. The categorical P4 verdicts of r15 and x1 were known. No tier-U unit had run.
- *Why.* The owner directed a relaxation so that the paper can report results over more cases.
- *Cost.* Tier U was defined after the categorical P4 verdicts of r15 and x1 were known. Its units are, by construction, ones that P4 did not find sufficient in every arm. So layer 5 is not an attributable three-arm comparison. It is stratified by the P4 pattern and reported apart.

The tier-U waves are:

- r15u: case-0614, 0615 and 0616, with their r15 cells;
- r18u: case-0502, 0507 and 0508, with their r18 cells;
- x1u: case-0604;
- x2u: case-0611.

F.4 Controller rulings that change how a rule is applied

DEV-S1: the S arm of record (2026-09-24, about 09:40Z).

- *What.* In the r15 origin run, every S attempt was refused, fail-closed, by the snapshot runner: the sealed evidence was not declared requalified under the executing revision. The ruling makes a separate S section the S arm of record. The S section runs the same S cells against requalified evidence. The origin S refusals remain in the record as refusals by design.
- *When.* During the r15 origin run, before any S run had produced output.
- *Why.* The readiness check did not test requalification.
- *Cost.* S no longer runs interleaved with L-fixed and L in the pre-declared block order. It runs at a different time and on a different machine. In every later wave the origin S intents are refused by the same design and are counted as failures of class D-S1.

R-P6X-SANITY: the omitted sanity check (2026-09-24, about 11:15Z).

- *What.* The controller's r15 prompt omitted the pre-registered technical sanity check of the exploratory models. The driver stopped, as designed, before any exploratory block. The ruling runs the check as frozen and resumes the exploratory blocks.
- *When.* Before any exploratory run.
- *Why.* It was a controller omission.
- *Cost.* The second exploratory model failed the check (provider failure) and is dropped. Its runs are skipped in every wave.

DEV-S2: the S-section clock (2026-09-24, about 13:30Z).

- *What.* The first S section (r16) was refused at the gateway on all twelve primary runs: the gateway cutoff did not precede its recovery time, because the runner used the wall clock rather than the capture's execution clock. The runner tooling was changed to use the execution-clock view. The second S section (r17) is the S arm of record for r15. The r16 section is superseded. It had also lacked the sanity records, a second controller omission.
- *When.* After r16 and before any S model call that produced output.
- *Why.* It was a readiness-method gap.
- *Cost.* r16 charged CNY 536.83 and measured none. A capture-start offset of three seconds between the S clock and the live capture is disclosed.

Screen v2 (2026-09-25, 03:47Z).

- *What.* The run-validity screen was restated as v2. Version v1 (03:34Z) was never used. v2 classifies runs whose subject returned no result from file names alone. It also admits two groups of runs that v1's wording would have excluded.
- *When.* After the r15 runs and before any analysis. It was written from run-directory file lists only.
- *Why.* v1 was worded against a file the runners do not write when the subject returns nothing.
- *Cost.* The screen was fixed after the r15 runs existed. It reads no outcome.

Screen gaps G1–G6 (2026-09-25, 08:41:40Z). Six gaps reported by the analysis builder were closed, including the conversion of partial scores to the reported scale. They were closed before any score was read.

Waves (2026-09-25, 10:34:23Z).

- *What.*
  - The screen and its gap rulings apply per wave.
  - Each wave's results are built in two steps: phase 1 (screens) is sealed before phase 2 (results).
  - Stop rule 6 is evaluated per wave.
  - The joint analysis runs once, at the earlier of two times: when every wave is done, or 2026-09-26T07:00Z. A wave not done by then is "not analysed (time)".
  - Intervals are reported only for a layer with at least eight clusters.
  - The exploratory models are reported per wave only, never pooled.
- *When.* Before any P6 score had been read. The r15 runs were done. No other wave had run.
- *Why.* The amendments had split the experiment into waves with different revisions and times.
- *Cost.* Waves are analysed on the clock. A slow wave may be left out of the joint analysis.

Smoke rulings for x2 (2026-09-25, 21:10:36Z). Three rulings, made before any x2 run, cover the real-host smoke test:

- a stub's exit status of 1 is accepted when the run's evaluator and execution records are clean;
- a compose image-exists build race gets one technical retry, in the smoke test only;
- each attempt uses its own smoke root.

They changed no P6 run.

Table 9: P6 execution record per wave, from the execution sentinels. C, F and S are runs completed, failed and skipped. Plan and C are given as P+X. F includes the origin D-S1 refusals (in parentheses), which are by design. No row states any outcome.

				Origin run				S section			
Wave	Status	Host	Units admitted (cases)	Plan	C	F	S	Plan	C	F	S
r15	partial	1	2 (0602, 0605)	36+36	14+4	42 (20)	12	12+12	7+6	7	4
r18	no P6: P4 NoGo	–	0 of 3			no run					
x1	no P6: P4 NoGo	–	0 of 1			no run					
x2	partial	2	2 (0501, 0613)	36+36	11+6	43 (20)	12	12+12	8+7	5	4
r15u	partial	1	3 (0614, 0615, 0616)	54+54	21+10	59 (30)	18	18+18	13+10	7	6
r18u	partial	1	3 (0502, 0507, 0508)	54+54	18+0	28 (16)	62	18+18	11+12	7	6
x1u	blocked at smoke	2	1 (0604)	18+18	0+0	0	0	6+6	0+0	0	0
x2u	partial (attempt b)	2	1 (0611)	18+18	8+3	19 (10)	6	6+6	4+2	4	2

r15 counts combine the origin run (2026-09-24T10:51:05Z) and its exploratory resume (13:16:27Z). Its S section is r17 (17:41:21Z). r18u is attempt e (2026-09-25T23:08:51Z), x2 is attempt f (23:46:57Z) and x1u is attempt a (2026-09-26T00:28:16Z). r15u finished at 2026-09-26T04:00:54Z. x2u ran twice. Attempt a (01:04:17Z) was blocked at the real-host smoke test before any P6 run: the worker-written smoke stub submitted a placeholder diagnosis that the model harness rejected locally, a stub defect; the host path (lifecycle, gateway, destroy) worked. Attempt b (04:03:15Z) ran with the controller-pinned stub of wave x2 (attempt f); its row is shown. The units that r18, x1 and x2 did not admit run in tier U (rows with suffix u).

F.5 P6 EXECUTION RECORD

Table 9 gives one row per wave. It reproduces the counts of each wave's execution sentinel. The origin run executes its plan on the host; the S section runs locally.

- P and X plans are 36 runs each per two units (three arms, three model slots, two replicates).
- Completed counts are split P+X.
- Failed counts include the origin S refusals of class D-S1, which are by design. Their number is given in parentheses.
- Skipped counts are exploratory runs of the dropped second slot, plus, for r18u, primary runs left unstarted by the disk-guard stop.

Failure classes by wave.

- *r15.* The primary origin failures are:
  - 12 D-S1 refusals;
  - 8 runs of one pre-registered model on the live arms that submitted no diagnosis;
  - 2 whole-call time refusals by the scorer, of one pre-registered model on case-0605.

  The exploratory origin failures are:
  - 8 D-S1 refusals;
  - 3 whole-call time refusals at finalization;
  - 8 invalid diagnoses;
  - 1 run with no diagnosis.

  The seven S-section failures are subject outcomes. The time refusals are reported as a missingness disclosure, under the rule fixed before any model run.
- *x2.* The 23 live failures are:
  - 9 of technical class: a compose-build race between concurrent live runs, all on case-0501;
  - 14 of subject class.

  The frozen rerun policy gives a budget of zero, so no run is rerun. The technical failures are reported as technical missingness.

- *r18u.* The 28 origin failures are 16 D-S1 refusals and 12 live failures.
- *r15u.* The 59 origin failures are 30 D-S1 refusals and 29 live failures; the S section has 7 failures.
- *x2u (attempt b).* The 19 origin failures are 10 D-S1 refusals and 9 live failures; the S section has 4 failures.

Spend. The execution sentinels record the following charged amounts, in CNY:

- r15 origin: 646.39;
- exploratory sanity check: 7.19;
- r15 exploratory resume: 501.86;
- superseded r16 section: 536.83;
- r17 section: 18.02;
- r18u: 752.16 (origin) and 32.05 (S section);
- x2: 1331.01 (origin) and 22.21 (S section);
- r15u: 1388.78 (origin) and 22.49 (S section);
- x2u: 0 (attempt a); 469.61 (origin) and 11.40 (S section) (attempt b);
- x1u: 0.

Charged amounts use a conservative per-run maximum where a run returns no token count. Their sum is CNY 5740.00, below the ceiling of stopping rule 1. No sentinel records a budget stop.

Superseded attempts. Some attempts were superseded before they touched a host, ran a unit or spent anything:

- *r18u attempts a–d.* They were blocked at plan construction. Attempt a failed the plan-count check. Attempt b had an unfilled placeholder. Attempt c had a units-schema key error. Attempt d had plan targets that did not match its units. Controller rulings on the plan and unit formats superseded them.
- *x2 attempts a–d.* Attempt a was stopped by the controller at step 1, because its prompt literal omitted a case. Attempts b–d failed as r18u's b–d did.
- *x2 attempt e.* It was blocked at the real-host smoke test, before any driver started. The L-fixed stub returned exit 1, which is by design. The L stub hit the compose build race. The smoke rulings above superseded it.
- *x2u attempt a.* It was blocked at the real-host smoke test, before any P6 run and without spend: its worker-written stub was defective (see Table 9). Attempt b superseded it.
- *The first P4 batch of x2.* It was voided before review, because of a template defect.

The r16 S section was superseded by r17 (DEV-S2). None of these choices depended on an outcome.

F.6 Exclusions and truncations

Every item below was decided before any analysis of the wave it affects.

- *r18 and x1: P4 NoGo.* No unit was admitted (0 of 3 on 2026-09-25 at 15:01:50Z; 0 of 1 at 12:45:30Z), so neither wave has a primary P6 run. These exclusions were decided before any model run on the units. Their units run in tier U (r18u, x1u).
- *x1u: blocked at the real-host smoke test.* This was on 2026-09-26 at 00:28:16Z, before any P6 run.
  - On case-0604, the live log tool returned an error after 30 s. Both smoke runs ended with an invalid diagnosis, although the same stub submission was accepted on the snapshot path.

  - A failing live log tool would degrade the evidence of every live-arm run and none of the snapshot runs.
  - The gateway is protected bytes, and the case environment is sealed.

  So x1u is blocked at the real-host smoke test before any P6 run: the live log tool returned 502 on case-0604, and the smoke stub of that generation was later found defective (x2u attempt a failed the same way with a worker-written stub), so a stub defect may have contributed; it was not re-attempted for lack of time. At 00:33:46Z the controller ruled that there would be no second attempt; at 01:13:58Z it recorded the stub finding and kept that ruling, deciding from the clock only (host 2 was expected to be held by x2u attempt b until the truncation; attempt b in fact finished before it, at 04:03:15Z). Both rulings were made before any x1u run existed. In the joint analysis, x1u has status "not started".

- *r18u: disk-guard stop.* The origin driver stopped at 2026-09-25T22:53:42Z (ledger sequence 129). A remote host-name readback came back empty, and the driver failed closed. This is a technical stop, not a cost or outcome stop.
  - Eight primary origin runs were left unstarted, and the origin exploratory phase was skipped.
  - The S section completed.

  At 2026-09-26T00:34:49Z the controller ruled that there would be no resume. It decided from the clock and the work involved only, before reading any output. The eight unstarted runs are reported as truncation.

- *x2: failures without rerun.* The nine technical failures and the fourteen subject failures stand as recorded. The rerun budget is zero, and no run is rerun.

- *D-S1 refusals.* Every origin S intent is refused by design (20 in r15, 16 in r18u, 20 in x2, 30 in r15u, 10 in x2u). The S arm of record is the S section.

- *Truncation.* The truncation time moved from 2026-09-26T02:00Z (Amendment 1) to 03:00Z (Amendment 5) and then to 05:00Z (Amendment 8). Each move was made before any P6 score was read. Blocks not started by 05:00Z are dropped whole and are reported as not run. None was dropped: r15u and x2u finished at 04:00:54Z and 04:03:15Z, and both sentinels record 0 dropped runs.

F.7 Disclosures of process lapses

- *x2 outcome-blindness finding.* The x2 producer parsed one run's finalization record once (case-0501, first replicate, L-fixed). It printed the record's top-level key names and three reason strings. It printed no score or diagnosis. The wave's outcomes were fixed before the read. Nothing was selected, excluded or rerun because of it. The controller did not read the file.

- *Controller self-report, 2026-09-25T12:47:26Z.* While recording the x1 P4 verdict, the controller printed the free-text reason field of the adjudication for case-0604. No run of that case existed, and no worker received the text. After this, the controller read only categorical P4 fields. Case-0604 is excluded anyway (x1 NoGo; x1u blocked at smoke).

- *Controller blindness slip, 2026-09-25T21:10:36Z.* While diagnosing x2 attempt e, a controller search printed infrastructure lines from the private reproduction directory of case-0501: image tags and a checksum list. It printed no ground truth, diagnosis or score. Later searches exclude the case directories.

- *Aggregate counts, 2026-09-25T22:43:36Z.* To classify the x2 technical failures, the controller read the run ledger's status and failure fields. For completed runs, the failed status correlates with a non-successful outcome. So the controller has seen aggregate completed and failed counts, which the workers' progress lines also show. No decision uses them.

- *x2u attempt a smoke log, 2026-09-26T01:13:58Z.* While diagnosing attempt a, the controller viewed part of its raw smoke log, which holds the outcome block of a stub run (stub provider, placeholder diagnosis; no real model, never scored or counted). Nothing was selected, excluded or rerun on it.

- *Local disk full, 2026-09-26 at about 04:12Z.* During the joint analysis, the local root file system filled with an editor's core dump, which the controller moved to an archive. The joint-analysis inputs were re-hashed and matched; the analysis completed with return code 0 and was not rerun.
- *r17 labels.* The r17 worker printed raw outcome labels for two runs of case-0602. Nothing was selected or rerun.
- *Hashing of private files.* A worker hashed three private specification files and printed only hash prefixes.
- *Controller omissions.* The exploratory sanity check was left out of the r15 prompt (R-P6X-SANITY). The sanity records were not carried into the r16 section (DEV-S2). Amendment 6 dropped three pre-registered cases (corrected by Amendment 7).
- *Controller stall.* No controller activity took place from 2026-09-24T17:49Z to 2026-09-25T03:27Z. Nothing was read in that interval.
- *x2 residue and log gap.* The x2 smoke test, and the smoke test of x2u attempt a (39 files), left root-owned files on host 2. They are left in place, because removal needs privileges the experiment does not use. The x2 progress-log gap (2026-09-25T21:45:16Z to 22:19:45Z) is a defect of the checker, not of the run.
- *Host side effects.* Every P6 execution and results sentinel, and the controller's own recounts, record zero files changed on the local interpreters or the package cache since the host baseline. The one earlier restoration the ledger mentions, of a package-owned interpreter file, predates the baseline and every P6 run. It does not affect P6.

F.8 What the deviations cost the inference

- *Clusters against the floor of eight.*
  - The primary tier holds four admitted units: case-0602 and 0605 (r15, two clusters, one application) and case-0501 and 0613 (x2, two clusters, one application). That is two applications, with 4 clusters having all three arms on J3, below the floor.
  - The layers as pre-registered, amended and extended, and their sensitivity layers, are therefore descriptive only. The amended layer and its sensitivity layer add no unit, because r18 admitted none.
  - Clusters with all three arms, from the joint analysis: J1 2, J3 4 and layer 5 (tier U with the primary units) 8. Layer 5 alone reaches the floor (status: reported).
- *Intervals.* Intervals are reported only for a layer that reaches the floor. Every other layer reports descriptive counts.
- *Tier U.* Tier-U units are analysed separately (J5), stratified by their P4 pattern, and never pooled into the primary layers. Their comparisons are not attributable in the sense of Section 5, because at least one of their cells was not found sufficient.
- *Selection and revisions.* Amendment 5 selects on a pre-model verdict. Tier U is defined after the categorical P4 verdicts were known. Units come from three execution revisions and several runner revisions, and the revision is confounded with application and mechanism. No selection uses a model outcome.
- *S timing.* The S arm of record runs in a separate local section, not in the block order with L-fixed and L (DEV-S1). A capture-start offset of three seconds is disclosed (DEV-S2).
- *Missingness.* Four sources remove observations; they are reported and never rerun:
  - technical failures (nine in x2);
  - the dropped exploratory slot;
  - the r18u truncation (eight primary runs and the whole origin exploratory phase);
  - whole-call time refusals.

  Only complete pairs are analysed.
- *Exploratory models.* They are described per wave only and never pooled across waves or with the pre-registered models.

- *Joint analysis.* It runs once, at the earlier of all waves done and 2026-09-26T07:00Z. It ran once, after the last wave's results were built, and finished at 2026-09-26T04:15:58Z; it was not rerun. Per-wave status: r15 (primary): analysed; r18 (primary): no_units; x1 (primary): no_units; x2 (primary): analysed; r15u (U): analysed; r18u (U): analysed; x1u (U): not_started; x2u (U): analysed.