ENVOS: AUDITING COMPUTER-USE AGENTS WHEN THE WORLD MOVES ON

Anonymous authors
Paper under double-blind review

ABSTRACT

Computer-use agents act on information that can become stale before they commit, but apparent failures under such change can also be produced by the evaluation itself. We introduce EnvOS, a protocol for auditing controlled mid-task state changes: changes fire only after the soon-stale value was served; feasibility, corruptions, isolation and delivery are checked; rules are frozen before rollout; separate code re-derives validity verdicts and the outcome terms supported by archived evidence. We evaluate the finalized protocol prospectively in two workflows under one controlled semantic-browser interface. In clinic booking, Claude Opus 5 succeeded on 0/16 instances with no panel, 2/16 with a cue-and-link panel and 15/16 when the same panel displayed the current location (LIVE–CUE: 13 rescues, 0 harms and 3 ties, block-local Holm $p = 0.000488$; CUE–OFF: block-local Holm $p = 0.5$). At the registered block-local level, displaying the current location increased success relative to this cue-and-link panel. The CUE–OFF contrast did not reject; this does not establish absence of a cue effect. Contradiction, salience, and reduced verification effort remain possible explanations. Over all twelve claim-bearing tests, LIVE–CUE's Holm $p = 0.00293$. In insurance-claim approval, Claude Opus 5 succeeded on 10/10 silent-change instances without review-step re-exposure, Claude Sonnet 5 on 1/10 without and 9/10 with it (8 rescues, 0 harms, 2 ties; registered block-local $p = 0.00781$; retrospective twelve-test Holm $p = 0.0781$). A separately pre-registered follow-up with a non-Anthropic configuration (Gemini 3.1 Pro Preview) did not meet the registered effect criterion (0/10 to 5/10, exact p = 0.0625). Evaluator audits exposed their own limits: on a sealed challenge, validity re-derivation met 24/28 expectations; on a second, specified without implementation or outcome access, the pipeline counted none of 26 corruptions as clean but gave no verdict on 2; on a third, which an external agent designed and generated, it caught more corruptions than raw-log baselines in both workflows. It still counted 12 of 39 as clean. These contrasts estimate within-configuration effects; unrejected comparisons do not establish absence or equivalence.

1 INTRODUCTION

Computer-use agents increasingly act in changing environments: a meeting moves, a record is superseded, or a recently read fact becomes stale. Recent benchmarks report frontier agents miss such updates and continue acting on superseded state (Yuan et al., 2026; Hui et al., 2026). Attributing such a failure to the change requires ruling out task, harness and scoring failures. A change can fire before the agent was served the old value; a harness can fail to provide working memory; a scorer can place the event one request early; a provider rate limit can be scored as a model decision. We met each in development (Section 6), so the evaluator itself must be tested.

This requires six checks: exposure before the change, a timely update of authoritative state, observation delivery, rejection of corrupted episodes, independent re-derivation of validity and endpoint terms, and interpretation fixed before rollout. To our knowledge, no computer-use evaluation establishes all six for the same controlled mid-task change before attributing a failure to it. Benchmarks such as OSWorld 2.0 and STT-Arena measure *whether* agents fail under change; EnvOS tests whether a measured failure can be *attributed* to the change.

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.

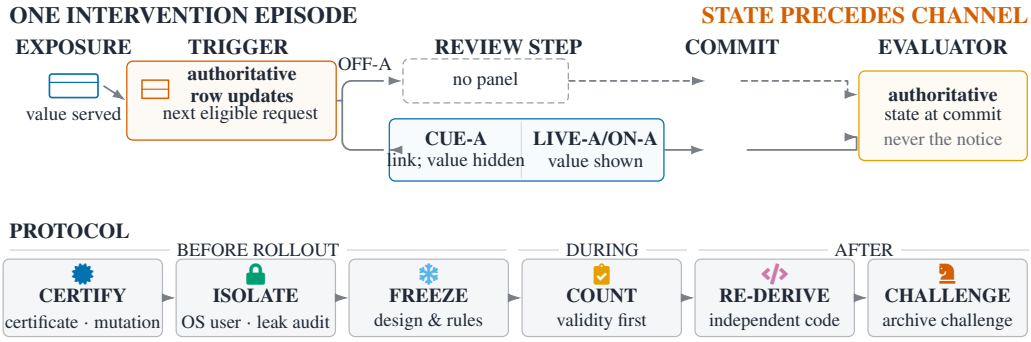

Figure 1: **EnvOS separates the authoritative state transition from what later reaches the agent** (Section 3). OFF-A has no review panel; CUE-A adds a source link with the value blanked, LIVE-A/ON-A the current value; the evaluator scores state at commit, never the notice. Lower band: the protocol stages (coverage and gaps: Table 9).

We introduce EnvOS, a protocol that makes these six checks separately testable (Figure 1), and apply it through a controlled semantic browser interface with a stateless per-step policy; reported effects are within-interface effects on the tested instances, not leaderboard scores or claims about native computer use. For each prospective block, the task definition, validity rules, analysis, code manifest, and interpretation templates were hash-frozen before the first rollout; later departures are reported rather than silently incorporated. Outcome interpretations are pre-written and selected mechanically. **Main results.** We evaluate the finalized protocol prospectively (Section 5). In clinic booking, Opus succeeded on 0/16 instances with no panel, 2/16 with a cue-and-link panel and 15/16 with the current value shown. The pre-registered primary contrast, value versus cue (LIVE–CUE), gives 13 rescues, 0 harms and 3 ties (block-local Holm $p = 0.000488$, paper-wide Holm $p = 0.00293$); cue versus no panel (CUE–OFF) has block-local Holm $p = 0.5$. At the registered block-local level, displaying the current location increased success relative to this cue-and-link panel. The CUE–OFF contrast did not reject; this does not establish absence of a cue effect. Contradiction, salience, and reduced verification effort remain possible explanations. In claims approval, Opus re-read the changed record in all 10 no-panel episodes and succeeded on 10/10; Sonnet re-read it in 1/10 and succeeded on 1/10 without versus 9/10 with review-step re-exposure (8 rescues, 0 harms, 2 ties; $p = 0.00781$): for Sonnet, the panel increased full-task success at the registered local level. That p is its registered block-local test; the twelve-test paper-wide Holm family, finalized after the claims outcomes, adjusts it to $p = 0.0781$. The instruction defines correctness by the policy value "as it stands when you approve," which may itself encourage checking. These are effects of what reaches the agent after a controlled change present in every compared arm, not a general effect of changing environments; exact p-values are fixed-instance, single-rollout quantities (Section 3.4). Our contributions are threefold. First, EnvOS, a bounded auditing protocol that makes these six checks separately testable (Section 3). Second, two prospectively frozen studies (Section 5): in clinic booking, at the block-local level the value display increased success relative to the cue-and-link panel (paper-wide Holm $p = 0.00293$); in a derived claims decision, for Sonnet 5 the panel increased full-task success at the registered local level (a block-local test; retrospective paper-wide Holm $p = 0.0781$), while Opus 5 succeeded on 10/10 no-panel instances. Third, auditing evidence: three archive challenges (two specified without access to the implementation or outcome tables, one also built by an external agent); post-hoc, author-built comparisons against simple raw-log checkers; and a defect taxonomy showing how evaluator errors can become model claims (Section 6). The anonymous supplement releases source code, frozen registrations, raw text archives, audit artifacts, and reproduction scripts.

2 Related Work

Agents under mid-task change. Recent benchmarks test agents under state, goal, and interface changes during execution. OSWorld 2.0 (Yuan et al., 2026) injects mid-task messages and a simulated user and reports that agents "miss information that arrives mid-task, guess rather than ask the user, and skip verification"; Gaia2 (Froger et al., 2026; 2025) lets asynchronous events modify state

while time flows; InterruptBench (Zou et al., 2026) revises the user's goal; and DynamicGUIBench (Liu et al., 2026), AgentHijack (Sun et al., 2026), and StressWeb (Bai et al., 2026) perturb the interface, perception, or execution, with StressWeb comparing against clean runs. STT-Arena (Hui et al., 2026) is closest in design: state-conditioned triggers can invalidate the agent's plan, and stale-state execution is a named failure mode. EnvOS fires a change only after the agent was served the value it supersedes, certifies that exposure from archives, and matches each changed arm with an inapplicable control sharing its trigger. This is a time-of-check-to-time-of-use gap (Lilienthal & Hong, 2025). Complementary work isolates neighboring pieces: STALE (Chao et al., 2026) tests noticing an invalidated memory without actions, EnvProbe (Song & Cai, 2026) tests when to re-query a belief, and Li & Shi (2026) studies richer between-step observation; our panel contrasts vary one review-step element before the commit.

Verifiable worlds and evaluation validity. A separate line of work asks whether the evaluator itself supports its verdict. OpenComputer (Wei et al., 2026) shows that hard-coded state verifiers agree with human adjudication better than a model judge on static tasks; ABC (Zhu et al., 2025) shows that task- and reward-design flaws can shift measured agent performance by up to 100% relative; and Kirgis et al. (2026) argue that outcome-only evaluation is not credible, documenting threats through log analysis. Related critiques question the construct and predictive validity of leaderboards (Zhu et al., 2026; Patel et al., 2026). EnvOS is a partial, executable response for dynamic interventions. Its methods (state-based scoring, pre-registration, mutation testing, separate implementations, provenance hashing) are established. EnvOS combines them around one intervention: a trigger that requires prior exposure, an inapplicable control sharing that trigger, and checks of the change's timing and delivery as well as the verdict. It archives 11 defects and reports the verdict terms it could not verify. Extended discussion and inherited conventions: Appendix X.

3 THE ENVOS PROTOCOL

EnvOS checks an evaluation before, during and after model rollouts. Each check verifies one limited property, not overall validity (gaps: Table 9 and §7). Figure 1 shows the stages, abstracted from the testbed in Appendix P. The six stages have been executed end-to-end, including independent audits, on both prospective workflows.

3.1 PHENOMENA AS REAL STATE TRANSITIONS AND EXPOSURE-CONDITIONED TRIGGERS

A phenomenon is a change to an application's authoritative (source-of-truth) state, not injected text, defined independently of any application and realized in each environment as a perturbation (an exploratory, non-preregistered clinic/hotel/travel corpus on the same scaffold illustrates this: Appendix V). Inside one of the agent's own requests, the application updates a portal row (a clinic session's location, a policy's limit) in one transaction and records the event. *State precedes channel*: a notice or panel renders only after the mutation, and the scorer judges the row at commit, never the notice.

Triggers are exposure-conditioned. The world records when it serves the page carrying the old value, then fires on the next eligible request, regardless of route. The archive shows that the page was requested before the event and appeared in a frame delivered to the policy; this certifies exposure, not comprehension. Each changed arm has an *inapplicable control*: the same trigger changes an irrelevant row, so the change occurs but the correct response is to change nothing.

An episode is scored by 5 invariants over world state and the agent's report: correct target state, exactly one commit, no collateral change, workflow continuation from the commit, and agreement between the report and committed record. Success requires every applicable invariant; the claims task has no step after the commit and omits continuation (independent verification: Table 9). Failures carry signatures, and checkpoints record world evidence separately from delivered frames, so delivery is never inferred from behaviour.

3.2 CHECKS, AND WHAT EACH VERIFIES

Before any model runs, each condition receives a *certificate* of machine checks. Every check is tri-state (PASS, VIOLATION, MONITOR ERROR); an unevaluable check is not a pass. World checks

confirm the arm fires exactly once, inside the anchored request and after exposure, with neutral work identical across arms. Discrimination checks run scripted policies in the real browser: the reference passes every arm and each declared deficient policy fails with its expected signature. *Checker checks* confirm the scorer independently locates the firing request.

A *mutation battery* corrupts the world or its record in a scripted reference episode. The checker must reject each corruption and accept positive controls. Before any model runs, a *per-instance feasibility* run confirms that the scripted policies discriminate on every frozen generated instance of each block.

An *independently implemented scorer*, written by the same team without shared evaluator code, re-derives the invariants from archived state; an unresolved term stays unresolved, never passed. A second implementation, written from the pre-registration text with only the standard library, re-derives the *validity verdicts* from raw logs; a prospective block gets this claim only once its own artifact exists, and in this build it is complete for BC, CLM-B and CLM-BS. A held-out mutation set was run once, sealed but not blind (Appendix C); Section 6 reports three run-once archive challenges of the auditing layers. 'Independent' means a separate code path (for challenge cases, another model agent), never an independent team.

3.3 Isolation and frame delivery

For finalized prospective blocks, the policy runs as a separate Unix user with a sealed argument list. It can read only the per-episode observation directory, has no path to the repository, run archives or evaluator, and navigates only task-declared origins. Observation delivery is recorded separately by channel. A planted-canary audit (5 canaries; 15 operating-system and 8 browser probes, plus a model probe; Appendix C) probes both boundaries. Historical predecessor blocks used earlier read and delivery rules (Appendix B); results of non-`Read` CLI tool calls were not archived (Section 7).

3.4 Freezing, counting, statistics and interpretation

Each block freezes its pre-registration (design, conditions, contrasts, test, budget, reserve, stopping rules, validity table, invariants and interpretation sentences) by sha256 before the first rollout. Its release manifest pins the sha256 of 29 repository paths, container images, seed generator, invoker and requested model id; the driver refuses any episode whose live digests differ. Post-freeze changes are dated amendments; predecessor blocks and departures are reconciled in Appendix B; each prospective analysis uses its block's frozen registration.

Prospective blocks count validity before outcome. An attempt excluded by the validity table, such as a provider rejection or a harness failure to supply the promised policy input, is replaced only under the predeclared reserve (2 per cell, 6 per block); every other valid attempt counts in its arm, including one that reaches the action budget. Only these validity failures are replaced (0 and 0 in the claims blocks). A counted clean-control failure stops the block as a harness finding: a stopping rule, not a diagnosis, since a valid clean task can fail by chance.

The statistical unit is a generated instance run under every arm of its block in a seeded order. Each registered paired contrast uses an exact two-sided sign test on discordant instance pairs at $\alpha = 0.05$ and reports treatment-only, control-only and tied pairs with a paired-difference interval (Newcombe, marginal 95 %). Arms follow a fixed counterbalanced order, not randomized assignment. Inference assumes exchangeability of discordant directions under the null and no carry-over or period effect.

Inference is conditional on these fixed instances and one sampled policy rollout per arm. Neither p-values nor intervals estimate variation across tasks or repeated model sampling; each contrast estimates a fixed-instance, within-interface total effect of assigning an arm. The within-order split is descriptive, with no prevalence claim. In CLM-BS the arm run first and the arm run second each succeeded on 5 of 10 pairs (by order: Appendix J); five pairs per order cannot exclude a period effect. Interpretation sentences are written before any outcome (Appendix K) and selected afterward by registered rules that are re-implemented and tested against the originals; a void, incomplete or inadmissible block contributes no inferential result (Appendix L). Each prospective block uses its pre-registered local test and multiplicity rule. Appendix L additionally reports a fixed twelve-slot paper-wide Holm sensitivity family. Because that family was finalized after the claims outcomes existed, its adjustment of the claims and historical tests is retrospective; BC's block-local Holm family was frozen before any BC outcome and is its prospective multiplicity control.

4 TWO WORKFLOWS, ONE HARNESS

World and harness. The shared testbed comprises 14 containerised workplace services: scaffold-generated portals for clinic booking, claims, procurement and inventory, with their databases, mail, calendar, chat, code hosting, task tracker, remote-controlled browser desktop and network proxy. The evaluated workflows use subsets of this infrastructure; the service count describes the shared testbed, not task breadth. Authoritative state is a row in an application's own database. Each episode is capped at 48 actions for clinic or 40 for claims. At every step, the harness archives a screenshot and text snapshot of the live page, requests one action from the sealed policy under a fixed schema, and executes it in the real browser. The policy is stateless between calls, so each task declares a memo channel: any action may carry at most 300 characters, and each prompt echoes the last 48 actions with their memos. Memos are archived, never scored, and equally available across a task's arms. Instances are generated deterministically from held-out test splits; all arms of an instance share identical task text, and instances are never pooled across models (generator and splits: Appendix Q).

The policy interface is a controlled semantic browser harness, not OSWorld-style native computer use (comparison with other harnesses: Appendix O). An arm's prefix names its review-step panel: OFF none, CUE a link to the source with the value blanked, ON (claims) and LIVE (clinic) the current authoritative value. Its suffix names what changes: -A the target record, -I an irrelevant one; in L0 nothing changes.

Claims approval: a derived decision. An agent approves a claim for the smaller of the eligible amount (claimed minus deductible) and the reimbursement limit on the claim's policy record as it stands at approval. The limit changes silently after the agent has been served the policy records page. ON-A re-renders the policy record, not the answer, at the review step, so the amount must still be derived (Figure 2); ON-I changes an irrelevant record and L0 changes nothing. ON-A−OFF-A therefore estimates the total effect of assigning the panel, bundling a verification cue, the current value, disagreement salience and saved lookup effort. Threshold-crossing instances ensure that copying the displayed policy limit can itself be wrong; the full numerical instance table and shortcut baseline are in Appendix J. The claims task shares the scaffold, harness, invoker and vendor with the clinic task: it is a second decision rule for the same intervention, not an independent environment, and no cue or announcement was varied in it.

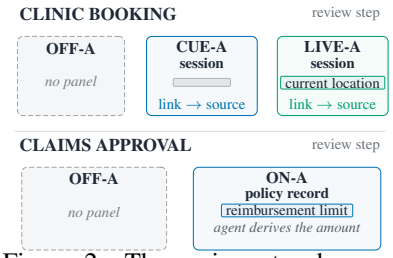

Figure 2: The review step by arm. OFF-A has no panel. Claims: ON-A shows the limit, not the derived answer. Clinic: CUE-A links to the source with location blanked; LIVE-A shows the current location.

Counting rules are registered per workflow: claims counts an untriggered episode as a failure, reported separately, under its own validity-first table; clinic blocks treat it as not an instance of the change and replace it (Appendices B and H).

Clinic booking: silent supersession, a cue or the value. Clinic booking combines information from mail, a calendar and the booking portal. The authoritative session location moves silently on the policy's first request to the booking wizard after the sessions page has been served. OFF-A has no panel. CUE-A adds a review-step panel naming the session and linking to its authoritative page with the location line blanked; LIVE-A is the same panel with the current location, so the arms differ only in that field (Figure 2). LIVE-I and OFF-I instead move an irrelevant session. Historical Block B used ON-A, a value panel from an earlier interface.

Certification requires a scripted link-following policy to pass CUE-A and LIVE-A but fail OFF-A, one revising only from a displayed value to pass LIVE-A but fail CUE-A, and one ignoring the panel to fail every target-change (-A) arm.

5 PROSPECTIVE RESULTS

Both prospective workflows were frozen before their first rollout, and their primary blocks completed under their registered gates. Each block is one model on one workflow: in claims approval

Table 1: Prospective model results. Success requires the full evaluator endpoint; in clinic rows its calendar-compatibility term rests on a stated design condition, not independent re-derivation. Paired counts: treatment-only/control-only/ties; p: exact two-sided sign test (BC: pre-registered block-Holm value in parentheses). 'indep. endpoint': counted rollouts whose independent reconstruction (clinic: without calendar compatibility) agrees with the evaluator. Blocks without results print their state; nets, intervals: Table 26. Gemini 3.1 Pro Preview received each frame inline in a stateless REST request with no tools; the Claude models were instructed to read each frame file with their file-read tool. Provider, model, invoker and image-delivery channel differ together; rows describe model–invoker configurations, not vendor effects.

	arm successes			paired	for/ag./	local p	indep.
model (block)	OFF	ON/CUE	LIVE	contrast	tie	(block Holm)	endpoint
claims approval							
Opus 5 (CLM-B)	10/10	10/10	—	ON−OFF	0/0/10	1.00	24/24
Sonnet 5 (CLM-BS)	1/10	9/10	—	ON−OFF	8/0/2	0.00781	24/24
Gemini (CLM-BG)	0/10	5/10	—	ON−OFF	5/0/5	0.0625	24/24
clinic booking							
Opus 5 (BC)	0/16	2/16	15/16	LIVE−CUE	13/0/3	0.000244 (0.000488)	51/51
				CUE−OFF	2/0/14	0.5 (0.5)	
Sonnet 5 (BCS)	*incomplete at the registered cutoff: 21 of 51 cells counted; no inference*						

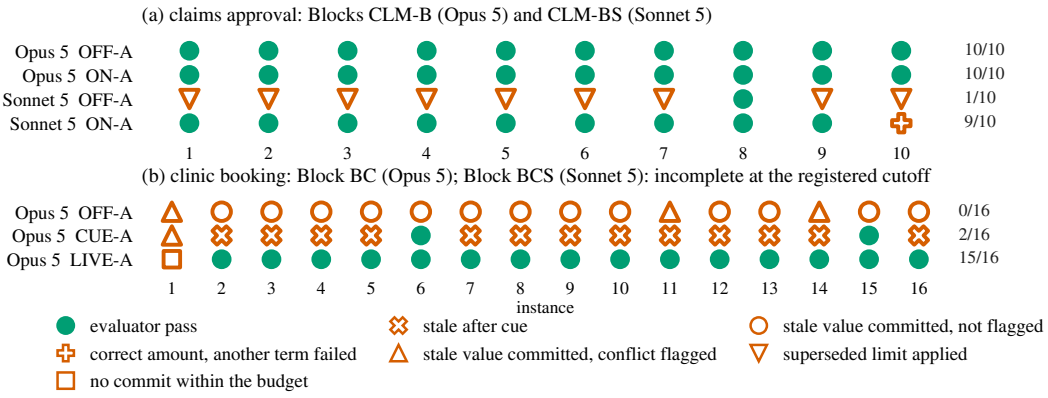

Figure 3: **Counted rollouts per paired instance.** (a) The Claude claims blocks (CLM-BG: Figure 4), without (OFF-A) and with (ON-A) the review-step policy panel, by model; (b) Block BC, no panel / cue-and-link / value panel (Figure 2). Columns are instances $1, \ldots, n$ (generator seeds: Appendices G–H); row totals at the right. Filled: evaluator pass; open: failure, shaped by signature.

CLM-B (Claude Opus 5) and CLM-BS (Claude Sonnet 5), with the Gemini follow-up CLM-BG; in clinic booking BC (Opus 5). BCS, a separately pre-registered Sonnet replication of BC on the same instances, was incomplete at its registered experiment cutoff, after a disclosed operator-caused halt (Appendix M), and contributes no inferential result. Table 1 summarizes the prospective blocks; Figure 3 shows paired-instance rollouts.

5.1 Claims approval: silent supersession in a derived decision

The claims blocks with Claude Opus 5 (CLM-B) and Claude Sonnet 5 (CLM-BS) were both registered before any claims rollout. Each pairs OFF-A with ON-A in one counterbalanced frozen order on the same 10 frozen generated instances (seeds 0–9), with ON-I and L0 controls; they differ only in the model and are never pooled. The decision function, gates and qualifiers are in Appendix H.

Sonnet (registered reading). Over 10 sealed instance pairs (seeds 0–9), Claude Sonnet 5 met the full approval endpoint on 1/10 without the panel and 9/10 with it; of the pairs, 8 favoured the panel, 0 the arm without it, and 2 neither; net $+0.80$ [0.37, 0.92] (marginal 95 % interval), $p = 0.00781$.

Controls: ON-I 3/3, L0 1/1. The review-step panel increased full-task success at the registered local level. The OFF-A failure categories and observed pre-commit revisions are reported separately (Appendix H). The OFF-A–ON-A contrast rejects under its registered block-local analysis but not under the paper-wide Holm analysis (adjusted $p = 0.0781$ over 12 tests). That family was finalized after the claims outcomes (Appendix L). The pre-registered derived-decision gate passed on all three threshold-crossing cases. Every outcome term the independent scorer re-derives is resolved and agreeing with the evaluator on 24 of 24 counted rollouts. The standard-library validity re-derivation agrees with the archive on 25 of 25 attempts. The independent scorer's scope covers all 4 registered outcome terms. Sonnet applied the superseded limit in 9/10 OFF-A episodes; in ON-A it requested the policy record after the change in 8/10. These process counts are descriptive (further counts: Appendix H).

Opus (full reading: Appendix H). Opus succeeded 10/10 in both OFF-A and ON-A; because every OFF-A instance already succeeded, the registered rescue-opportunity gate was not met and no panel-effect claim is made for Opus. Opus requested the policy record after the change in all 10 OFF-A runs, versus 1/10 for Sonnet; these process counts are descriptive, and the instruction itself may encourage checking.

Gemini follow-up (CLM-BG). On the ten registered reused claims instances, under the frozen Gemini 3.1 Pro Preview configuration, OFF-A failed on 10/10 instances, but review-step re-exposure did not meet the registered effect criterion (5 rescues, 0 harms, 5 ties; exact paired p = 0.0625); the effect is not shown for these instances under this configuration at this budget, and this is not evidence of no effect.

5.2 Clinic booking: a verification cue versus the value

Historical Block B's value panel (Appendix T) did two things at the review step: it cued a re-check of the session and delivered the value. The clinic block with Opus 5 (BC) separates the two. Each of 16 fresh instances runs with no panel (OFF-A), the cue-and-link panel (CUE-A) and the value panel (LIVE-A) in a frozen counterbalanced order, beside 3 LIVE-I controls. The primary contrast, LIVE–CUE, asks whether showing the value changes success; the key secondary, CUE–OFF, tests the cue-and-link panel against no panel (unconditionally). Both are Holm-adjusted within the block. The estimand is the total effect at the fixed 48-action budget, including the navigation the value saves. The design was enlarged before any BC rollout (Appendix M).

Result. Claude Opus 5 succeeded on 0/16 OFF-A, 2/16 CUE-A and 15/16 LIVE-A instances. LIVE–CUE yielded 13 rescues, 0 harms and 3 ties (net $+0.81$ [0.49, 0.92], exact $p = 0.000244$; block-local Holm $p = 0.000488$). CUE–OFF yielded 2 rescues, 0 harms and 14 ties (net $+0.13$ [-0.09, 0.36], exact $p = 0.5$; block-local Holm $p = 0.5$). Only 1 of 16 CUE-A runs followed the panel's link (descriptive). At the registered block-local level, displaying the current location increased success relative to this cue-and-link panel. The CUE–OFF contrast did not reject; this does not establish absence of a cue effect. Contradiction, salience, and reduced verification effort remain possible explanations. The LIVE-I controls passed 3/3; nothing is attributed to the panel from unmatched controls. The independent scorer reconstructs appointment availability from archived rows; calendar compatibility is not re-derived but rests on stated design conditions (the generator and state preservation), whose checks are reported separately. Under that condition, every outcome term is resolved and agreeing with the evaluator on 51 of 51 counted rollouts (Table 9). Independent validity re-derivation agrees with the archive on 53 of 53 attempts. The LIVE–CUE rejection survives the paper-wide Holm analysis over 12 claim-bearing tests (adjusted $p = 0.00293$).

6 Auditing the evaluator: challenges, baselines, defects

Sealed challenge. A separate model built a run-once 28-case challenge, white-box to its builder but unseen by the Claude agents that built the pipeline; the outcome scorer met 28/28 sealed expectations, validity re-derivation 24/28 and archived evaluator replay 10/28, unrepaired (Table 7, Appendix C); not blind validation.

Table 2: **The protocol against a simple raw-log baseline** (post hoc, author-built, not blind; separate from Table 7). Same raw evidence and scored properties. Controls accepted; corruptions rejected (not counted, or counted with the corrupted invariant flagged), in brackets those in the baseline's declared scope. Clinic protocol rows are derived from the sealed tools' recorded outputs; the claims pipeline voids `C09b` as invalid rather than failing its single-commit term. Every case: Appendix D.

checker	controls	corruptions (in scope)	would enter the results unflagged
clinic: the sealed challenge's 10 controls and 18 corruptions, its recorded outputs re-tabulated			
protocol, implemented reading	10/10	15/18 (11/12)	notice delivery (`C01`); panel delivery (`C02`); commit before change (`C08`)
protocol, registered reading	10/10	16/18 (11/12)	commit before change (`C08`); truncated trajectory (`C15`)
simple baseline (143 lines)	9/10	11/18 (11/12)	7, of which 6 outside its declared scope
claims: 10 controls and 19 corruptions we built from counted archives			
operational pipeline	10/10	16/19 (10/12)	duplicate commit in the access log only (`C09a`); truncated trajectory (`C15`); stale pages after the change (`C17p`)
simple baseline (108 lines)	10/10	6/19 (6/12)	13, of which 7 outside its declared scope

Rule-only case-design challenge. A fresh GPT-6 chat given only rule excerpts and the archive schema (no code, paper, earlier cases or outcomes) specified 56 cases with expected verdicts. We implemented them on archive copies under frozen rules; a second chat reviewed fidelity, and 36 admitted cases ran once. The pipeline accepted all 10 controls and, of 26 corruptions, caught 24 (one under another class), gave no verdict on 2 and counted none as clean (outcome scorer alone: 10; raw-log baselines: 19). Only the case design was blind, and clinic coverage is thin (Appendix E).

External challenge. A Gemini CLI agent given only the rules, schema and 15 archive copies designed 56 cases *and* wrote their generator, which we ran unread, sealed and scored once per tool. The pipeline caught 14/20 claims and 11/19 clinic corruptions (raw-log baselines 1, 3), counted 4 and 8 as clean (baselines 15, 15) and rejected 3 of 16 controls (baselines $\leq$1). We dispute 19 expectations; it used one base per workflow, and an LLM agent is no independent team (Appendix F).

Against a simple raw-log baseline. This comparison is post hoc, author-built and not blind: after the sealed result, which stands unchanged, we wrote a simple checker rebuilding each verdict from the same raw state, access log and served pages, with the pipeline in view, and sealed each before its one run (Table 2). No protocol path rejected a control; the clinic baseline rejected one. Within the baseline's declared scope both rejected 11 of 12 clinic corruptions; on claims the pipeline rejected 10 and the baseline 6 of 12. A careful simple checker thus reproduced most narrow clinic checks; EnvOS adds breadth (delivery, served arm, collateral state) that such a checker could be extended to cover, and the baseline alone caught one case (Appendix D). As we built both cases and checkers, only the external challenge above tests independent construction, and there the pipeline counted 4 of 20 claims and 8 of 19 clinic corruptions as clean.

Development defects are not a detection rate. Certification, audits, and re-derivation surfaced 11 defects; each is now a regression check. For 7, an archived row scored under the defect already carried a model verdict (Table 34, Appendix W). Each block is interpreted only under the code that governed it.

7 Limitations and conclusion

Scope and power. The two prospective workflows share one harness and scaffold; we claim reuse across authoritative objects and decision rules, not cross-environment generality. The harness is a controlled semantic browser with a stateless per-step policy and a self-authored memo channel; the policies are Claude models, plus one Gemini configuration in claims only. All prospective inferential claims of the paper rest on Claude-model blocks; the only non-Anthropic block (CLM-BG) is a separately registered configuration follow-up on reused instances. Cross-configuration differences

do not isolate vendor or model effects. A pre-registered screenshot-only coordinate-action crossover of the claims workflow stopped at its engineering admission gate before any model rollout: viewport fit failed (18 of 144 page checks did not fit) and legibility failed (4 of 404 values unrecovered at 1536×864). It therefore contributes no behavioral evidence, and interface robustness remains untested. Each claims block has 10 pairs, one run per cell, so repeatability on the same instances is not assessed. The intervention deliberately targets the gap between checking and using a value, so we estimate effects at that adversarial timing, not a general propensity to fail under environmental change. Both paired arms contain the change, and the ON-I and L0 controls do not supply a matched no-panel comparison, so the effect of the change itself is not estimated. At $n = 16$, BC resolves only large paired effects; non-rejection establishes neither absence nor equivalence.

What the checks cover, and what they do not. The claims blocks verify every outcome term; the prospective clinic blocks archive the appointment rows the nearest-slot term needs, while calendar compatibility rests on stated design conditions; the historical blocks verify the decision term on every booking and no success in full (Table 9). Not covered: comprehension (checkpoints certify delivery); paths no scripted policy takes, where several defects lay; the CLI's system prompt and decoding defaults (Appendix R). The Claude CLI could emit tool calls besides the allowed frame read, equally available across paired arms and denied evaluator access by OS isolation; their provider-side results were not archived, so policy-interface reconstruction is incomplete (names and counts: Appendices J and R). The held-out mutation set was not blind, and the raw-log baseline comparisons are post hoc and author-built (most corruptions from one archive: 14/18 clinic, 16/19 claims), so incremental coverage is case by case, not a rate; the reconstruction and repaired checker were written knowing the fault classes and read the same archives: independent code, not independent evidence.

Conclusion. The prospective claims and clinic experiments, not the historical predecessor blocks, carry the model-effect conclusions (Section 5). In clinic booking, at the block-local level the value display increased success relative to the cue-and-link panel (LIVE–CUE block-local Holm $p = 0.000488$, paper-wide $p = 0.00293$; CUE–OFF block-local $p = 0.5$). In claims approval Opus succeeded 10/10 without review-step re-exposure, and for Sonnet the panel increased full-task success at the registered local level (block-local $p = 0.00781$; retrospective paper-wide Holm $p = 0.0781$). These contrasts estimate within-configuration effects; unrejected comparisons do not establish absence or equivalence. The auditing challenges and the post-hoc, author-built raw-log comparisons show concretely that intervention, delivery and scoring checks can disagree on the same corrupted archive, so each layer must be tested rather than assumed correct.

Reproducibility statement

Every quantitative statement in this paper is a macro emitted by a claim ledger (`claims.py`) that re-derives it from a hashed artifact and fails on any missing or drifting input and on any macro used but not defined (Appendix Y); every sentence interpreting a prospective block is a template selected by the ledger (Appendix K); ADDENDA 18 and 19 and the claims pre-registration are frozen (each file verifies against its sha256 record) and the interpretation templates are sealed. The figures are generated from the same artifacts. The anonymous supplement contains the environment, task-module, certifier, scorer, harness, invoker and test sources; the pre-registrations, certificates, block specifications, release manifests and search logs; the raw text archives of the historical, claims (CLM-BG included) and Blocks BC and BCS model episodes (no screenshot frame ships, for size); the held-out battery and the sealed archive challenge after its one run; and a reproduction script that needs no network, world or model. The anonymous supplement includes the raw text archives, search logs, results, independent scorer outputs and validity re-derivations for every prospective block reported in the main text, including BC; a no-model reproduction target recomputes the cells of Table 1, the per-instance grids of Figure 3, all registered paired tests and the associated audit counts from those packaged artifacts. Evidence that cannot be reconstructed, such as slot occupancy at booking time in the historical blocks and the CLI's unarchived defaults, is listed rather than inferred: Appendix Y states the result of an offline run of the supplement's reproduction script and what is permanently missing, and points to the supplement's complete inventory, which ships as a generated provenance file.

ETHICS STATEMENT

The testbed is synthetic: every person, appointment, claim, message and record in the office-v1 world is generated, and no human subjects, personal data or production systems are involved. The evaluated policies are commercial Claude models accessed through the vendor's command-line interface (the Claude CLI) and, in Block CLM-BG, Gemini 3.1 Pro Preview accessed through Google's REST API, under their providers' terms; the isolation described in §3.3 exists to protect the evaluation, not to circumvent any provider control. The paper reports a non-replication of a result we published ourselves; the earlier evidence is preserved and cited as superseded, not withdrawn.

AI USE STATEMENT

We disclose LLM involvement against the categories ICLR asks about. **Used:** method implementation (Claude Code was used to author task modules, checkers and analysis code, all of which is covered by offline tests, certification and the independent scorer); research methodology (GPT-6 was consulted on the experimental design, the pre-registration addenda, the sizing of the blocks and the interpretation templates; every consult, answer and disposition is archived); evaluation infrastructure (GPT-6, through the Codex CLI, built the second sealed archive challenge of Section 6, which no Claude agent read before its run; in a web chat, GPT-6 specified the cases and expected verdicts of the rule-only case-design challenge of Section 6, and a second chat reviewed the fidelity of the authors' implementation of each case; Appendix E; a Gemini CLI agent (`gemini-3.1-pro-preview`) designed the cases and expected verdicts of the externally built challenge and wrote the generator that instantiated them, Appendix F); writing (GPT-6 critiqued drafts of the abstract and of this manuscript; Claude Code drafted sections from the project's own documents). **LLMs as the object of study:** the evaluated policies are LLM-based computer-use agents: Claude models and, in the claims block CLM-BG only, Gemini 3.1 Pro Preview (`gemini-3.1-pro-preview`); no GPT-6 model result is reported (Appendix M). **Not used:** no measurement in this paper was produced by an LLM; every number is derived mechanically from archived artifacts by the claim ledger, and no experimental outcome was generated, altered or selected by an LLM. The research question, the decision of what to claim, and every acceptance of a result were the authors'. The authors take full responsibility for all content, including any error introduced this way.

REFERENCES

Haoyue Bai, Dong Wang, Long Chen, Bingguang Hao, Pengyang Shao, Yonghui Yang, Yicheng He, and Chenyi Zhuang. StressWeb: A diagnostic benchmark for web agent robustness under realistic interaction variability. *arXiv preprint arXiv:2604.16385*, 2026. URL https://arxiv.org/abs/2604.16385.

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

Hanxiang Chao, Yihan Bai, Rui Sheng, Tianle Li, and Yushi Sun. STALE: Can LLM agents know when their memories are no longer valid? *arXiv preprint arXiv:2605.06527*, 2026. URL https://arxiv.org/abs/2605.06527.

Romain Froger, Pierre Andrews, Matteo Bettini, Amar Budhiraja, Ricardo Silveira Cabral, Virginie Do, Emilien Garreau, Jean-Baptiste Gaya, et al. ARE: Scaling up agent environments and evaluations. *arXiv preprint arXiv:2509.17158*, 2025. URL https://arxiv.org/abs/2509.17158.

Romain Froger, Pierre Andrews, Matteo Bettini, Amar Budhiraja, Ricardo Silveira Cabral, Virginie Do, Emilien Garreau, Jean-Baptiste Gaya, Hugo Laurençon, Maxime Lecanu, Kunal Malkan, Dheeraj Mekala, Pierre Ménard, Gerard Moreno-Torres Bertran, Ulyana Piterbarg, Mikhail Plekhanov, Mathieu Rita, Andrey Rusakov, Vladislav Vorotilov, Mengjue Wang, Ian Yu, Amine Benhalloum, Grégoire Mialon, and Thomas Scialom. Gaia2: Benchmarking LLM agents on dynamic and asynchronous environments. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=9gw03JpKK4.

Tingfeng Hui, Hao Xu, Pengyu Zhu, Hongsheng Xin, Kun Zhan, Sen Su, Chunxiao Liu, and Ning Miao. STT-Arena: A more realistic environment for tool-using with spatio-temporal dynamics. *arXiv preprint arXiv:2605.18548*, 2026. URL https://arxiv.org/abs/2605.18548.

Peter Kirgis, Sayash Kapoor, Stephan Rabanser, Nitya Nadgir, Cozmin Ududec, Magda Dubois, JJ Allaire, Conrad Stosz, Marius Hobbhahn, Jacob Steinhardt, and Arvind Narayanan. Log analysis is necessary for credible evaluation of AI agents. *arXiv preprint arXiv:2605.08545*, 2026. URL https://arxiv.org/abs/2605.08545.

Bojie Li and Noah Shi. Agent-computer observation interfaces enable dynamic computer use. *arXiv preprint arXiv:2606.29472*, 2026. URL https://arxiv.org/abs/2606.29472.

Derek Lilienthal and Sanghyun Hong. Mind the gap: Time-of-check to time-of-use vulnerabilities in LLM-enabled agents. *arXiv preprint arXiv:2508.17155*, 2025. URL https://arxiv.org/abs/2508.17155.

Enqi Liu, Liyuan Pan, Zhi Gao, Yan Yang, Chenrui Shi, Yang Liu, Jingrong Wu, and Qing Li. Benchmarking and improving GUI agents in high-dynamic environments. *arXiv preprint arXiv:2604.25380*, 2026. URL https://arxiv.org/abs/2604.25380.

Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947. doi: 10.1007/BF02295996.

Dhaval C. Patel, Kaoutar El Maghraoui, Shuxin Lin, Yusheng Li, Tianjun Feng, Chun-Yi Tsai, et al. Beyond static leaderboards: Predictive validity for the evaluation of LLM agents. *arXiv preprint arXiv:2606.19704*, 2026. URL https://arxiv.org/abs/2606.19704.

Xinyuan Song and Zekun Cai. Ask the world before acting: Environment probing for calibrated agent world models. *arXiv preprint arXiv:2606.31422*, 2026. URL https://arxiv.org/abs/2606.31422.

Jingwei Sun, Jianing Zhu, Yuanyi Li, Tongliang Liu, Xia Hu, and Bo Han. AgentHijack: Benchmarking computer use agent robustness to common environment corruptions. In *International Conference on Machine Learning (ICML)*, 2026. URL https://arxiv.org/abs/2605.25707.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. AppWorld: A controllable world of apps and people for benchmarking interactive coding agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16022–16076, 2024. doi: 10.18653/v1/2024.acl-long.850. URL https://aclanthology.org/2024.acl-long.850/.

Jinbiao Wei, Qianran Ma, Yilun Zhao, Xiao Zhou, Kangqi Ni, Guo Gan, and Arman Cohan. OpenComputer: Verifiable software worlds for computer-use agents. *arXiv preprint arXiv:2605.19769*, 2026. URL https://arxiv.org/abs/2605.19769.

Yule Wen, Yixin Ye, Yanzhe Zhang, Diyi Yang, and Hao Zhu. Real-time reasoning agents in evolving environments. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2511.04898.

Edwin B. Wilson. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212, 1927. doi: 10.1080/01621459.1927.10502953.

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts. In *International Conference on Learning Representations (ICLR)*, 2024a. URL https://arxiv.org/abs/2305.13300.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2024b. URL https://arxiv.org/abs/2404.07972.

Frank F. Xu, Yufan Song, Boxuan Li, Yuxuan Tang, Kritanjali Jain, Mengxue Bao, Zora Z. Wang, Xuhui Zhou, Zhitong Guo, Murong Cao, et al. TheAgentCompany: Benchmarking LLM agents on consequential real world tasks. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2025. URL https://papers.nips.cc/paper_files/paper/2025/hash/0d744742f6fac4d1134c019b7cef3c8a-Abstract-Datasets_and_Benchmarks_Track.html.

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

Mengqi Yuan, Zilong Zhou, Xinzhuang Xiong, Weiming Wu, Jiayang Sun, Jiamin Song, Kaiqian Cui, Bowen Wang, Haoyuan Wu, Yitong Li, Dunjie Lu, Haikong Lu, Qi Zhen, Xinyuan Wang, Jiaqi Deng, Yuhao Yang, Cheng Chen, Boyuan Zheng, Alex Su, Xiao Yu, Hao Zou, Saaket Agashe, Xing Han Lu, Manpreet Kaur, Zhengyang Qi, Vincent Sunn Chen, Frederic Sala, Dayiheng Liu, Junyang Lin, Zhou Yu, Yu Su, Siva Reddy, Xin Eric Wang, Peng Qi, Tianbao Xie, and Tao Yu. OSWorld 2.0: Benchmarking computer use agents on long-horizon real-world tasks. *arXiv preprint arXiv:2606.29537*, 2026. URL https://arxiv.org/abs/2606.29537.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. WebArena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2307.13854.

Pengyu Zhu, Li Sun, Philip S. Yu, and Sen Su. "LLM agent performance" is not a single evaluation target. *arXiv preprint arXiv:2602.03238*, 2026. URL https://arxiv.org/abs/2602.03238.

Yuxuan Zhu, Tengjun Jin, Yada Pruksachatkun, Andy Zhang, Shu Liu, Sasha Cui, Sayash Kapoor, Shayne Longpre, Kevin Meng, Rebecca Weiss, et al. Establishing best practices in building rigorous agentic benchmarks. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2025. URL https://arxiv.org/abs/2507.02825.

Henry Peng Zou, Chunyu Miao, Wei-Chieh Huang, Yankai Chen, Yue Zhou, Hanrong Zhang, Yaozu Wu, Liancheng Fang, Zhengyao Gu, Zhen Zhang, Kening Zheng, Fangxin Wang, Yi Nian, Shanghao Li, Wenzhe Fan, Langzhou He, Weizhi Zhang, Xue Liu, and Philip S. Yu. When users change their mind: Evaluating interruptible agents in long-horizon web navigation. *arXiv preprint arXiv:2604.00892*, 2026. URL https://arxiv.org/abs/2604.00892.

The appendices put the final evidence first: the validity table and the rule reconciliation (Appendices A–B), the certificates, the scorer's coverage and the independent re-derivations (Appendix C), then the prospective blocks—their records, the verbatim interpretation templates, the paper-wide multiplicity family, the decisions of the final night and a diagnostic invariant-completion fraction of the claims blocks (Appendices G–N)—then specification and per-episode records of the historical, hypothesis-generating blocks; the registration chronology with its errata follows them (Appendix U). The frozen pre-registrations, the release manifests and the full claim ledger are files, not typeset here (Appendices U and Y).

A VALIDITY TABLE

Section 2 of the pre-registration, generated from the frozen file. 7 verdicts are not counted and are replaced under the declared reserve; 4 outcomes are counted. The historical blocks ran success-first; only such validity failures were replaced there: four historically (A2 1, BS 3; each independently

re-derived as a harness error). Under the supersession family's pre-registration a contrast whose control cell falls below 6/8 is not headlined (moved from Section 3.4).

event	verdict	counted	replaced
provider/CLI rejects before any action	`infra_rejected`	no	yes (fixed rule above)
action raised / app unreadable	`infra_failure`	no	yes
scorer MONITOR_ERROR	`verifier_failure`	no	yes, after the checker is fixed and re-certified
arm served $\neq$ rung requested	`arm_mismatch`	no	yes
injector fired outside its anchor	`invalid_intervention`	no	yes
event log fired but the session row carries no ΔS	`state_delta_lost`	no	yes
notice on screen but frame not delivered	`observation_not_delivered`	no	yes
agent reaches the 48-action budget without success	`budget_limited`	yes (failure)	no
agent never opens an available notice (email arms only; n/a here)	`notice_not_opened`	yes	no
control fails for a behavioural reason	`model_failure`	yes	no
success	`success`	yes	—

B REGISTERED VERSUS IMPLEMENTED RULES

Every validity or analysis rule, where the pre-registration registers it (line of the frozen file), where the code implements it (file and line; the pre-image where the code later changed), the dated correction that followed, and which archived values that correction re-derived. The five current model-block logs hold 82 attempts and 78 counted rollouts, each attempt matched one to one to a run directory (82). Generated by `audit/rule_reconciliation.py`, whose independent re-derivations of the delivery checkpoints agree with the archived values on every row.

rule	blocks	registered	implemented	retrospective correction	archived values re-derived
Trigger anchor	A (superseded)	PRECOMMIT:19	anchor 'second authenticated GET /' (code replaced; archived in superseded log rows)	ADD 3 (PRECOMMIT:203): exposure-conditioned anchor for new blocks; scorer after_login fix (PRECOMMIT:213)	none persisted: archived C4 of superseded S-L2/ S-C s0-s2 still reads False; 3 untriggered episodes stay uncounted; block superseded
Trigger anchor	A2, A2S	PRECOMMIT:206	superseded_session.py:442	none	none
Trigger anchor	B, BS, BF	PRECOMMIT:274	superseded_session.py:449	ADD 5 kept the late fire and redefined the reference policy (PRECOMMIT:280) – before any rollout	none (prospective)
Validity before outcome (ladder order)	all	PRECOMMIT:41	telemetry.py:32 (doc) vs telemetry.py:232 precedes telemetry.py:239	ADD 11 discloses success-first order (PRECOMMIT:454); ADD 16 repeats it (PRECOMMIT:653)	no verdict re-derived; this script lists SUCCESS rows that validity-first would void
Silent-arm perception branch	B (row 1); BS, BF prospective	PRECOMMIT:42-54 (the §2 table has no silent-arm row)	telemetry.py:253-254	ADD 10 (PRECOMMIT:401)	changed_FRX-OFF-A_test_id_s20_claude-opus-5: delivery_ambiguous $\rightarrow$ model_failure; pre-fix classifier would have voided 25 silent-arm model failures

rule	blocks	registered	implemented	retrospective correction	archived values re-derived
Notice delivery (C3)	A2, A2S (announced arms)	PRECOMMIT:50	preimage[2026-09-22T0004]/ provenance.py:632 (pre-image; frame + ack on classified route) → provenance.py:663; verdict telemetry.py:266	ADD 11 (ack on unclassified route, PRECOMMIT:438); ADD 16 (text or image, PRECOMMIT:619)	ADD 11: 5 A2 rows re-derived (changed_ S-L2_s9 opus-5, changed_S-L2_s11 opus-5, cosmetic_S-C_ s13 opus-5, cosmetic_ S-C_s14 opus-5, cosmetic_S-C_s15 opus-5): C3 false→true on 4, 4 signature changes; ADD 16: C3 false→true on 2 rows (cosmetic_ S-C_s12 opus-5, cosmetic_S-C_s15 opus-5); 0 verdict changes
Re-exposure panel delivery	B, BS, BF (FRX-ON-*)	PRECOMMIT:305-308	preimage[2026-09-22T1437]/superseded_ session.py:1934 (pre-image: screenshot required, checkpoint + signature only); never a verdict: telemetry.py:254 reads C3 only	ADD 16 redefines delivery as text or image after BS outcomes (PRECOMMIT:619); ADD 17 defect B discloses non-enforcement (PRECOMMIT:733)	ADD 16: panel checkpoint false→true on 5 rows (changed_ FRX-ON-A_s22 sonnet-5, changed_ FRX-ON-A_s27 sonnet-5, changed_ FRX-ON-A_s28 sonnet-5, cosmetic_ FRX-ON-I_s21 sonnet-5, cosmetic_FRX-ON-I_s22 sonnet-5); signatures of BS ON-A s27, s28 STALE_APPLIED_ UNFLAGGED → STALE_AFTER_ REEXPOSURE; 0 verdict changes; ADD 17: none
Read-target (leak) rule	all (voids: A2 1, BS 3)	not in the §2 table; harness rule (PRE-COMMIT:508)	web_rollout.py:910; override web_ rollout.py:1942	ADD 12 (PRECOMMIT:495): errored reads and /dev/ null no longer void	none: the 4 voided attempts stay uncounted (harness stopped them); cells re-run as replacements
Action budget	all	PRECOMMIT:51	telemetry.py:218	none	none
Provider limit	A (superseded), all later	PRECOMMIT:149	telemetry.py:157	none (ADD 2 predates every counted block)	none
Registered claim rule (all kept failures STALE_ APPLIED_ UNFLAGGED)	B, BS, BF, A2	PRECOMMIT:89-90	superseded_ session.py:1032; superseded_ session.py:2200	ADD 17 defect A discloses the distractor-clinic flag (PRECOMMIT:705)	none: labels kept; §6 rule NOT met for Block B (3 FLAGGED OFF-A rows)
Replacement reserve / stop rules	all	PRECOMMIT:35-39	run_block.py:97	ADD 12 resumed BS after one repair cycle as registered	none
Primary contrast	A2 / B / BS; A2S, BF descriptive	PRECOMMIT:26; B PRECOMMIT:287; BS PRECOMMIT:349; A2S/BF PRECOMMIT:537	stats.py:43	ADD 13 withdrew A2S's test and downsized A2S/BF before their rollouts	none

What the corrections changed. ADDENDUM 10 re-derived one Block B row; the classifier before it would have voided 25 model failures of the silent blocks, those of the re-exposure arm included (every arm without a notice, the sense of "silent-arm" in the rule matrix above). ADDENDUM 11 re-derived 5 Block A2 rows and changed 4 signatures; ADDENDUM 16's delivery redefinition flipped 7 checkpoints and changed 4 signatures; neither changed a verdict. The implemented classifier tested success before delivery, so a success was never checked for validity; that later corrections left the verdicts unchanged shows what re-ordering changes on these archives, not that the success-first procedure was valid. The validity-first analysis covers 53 success rows: the 46 counted successes of the five current blocks and the 7 counted successes of the superseded Block A restart (ADDENDUM 3), which `rule_reconciliation.py` examines alongside them. Of these, 9 would have been voided under a validity-first order with some archived version of the delivery checkpoint: 8 current rows (A2 S-L2 s9; A2 S-L2 s11; A2 S-C s12; A2 S-C s13; A2 S-C s14; A2 S-C s15; BS FRX-ON-I s21; BS FRX-ON-I s22) and 1 of the superseded block.

A manifest rebuilt mid-block. A dated amendment can rebuild a block's manifest mid-block (ADDENDA 11 and 12: 4 Block BS attempts, 1 counted, ran under the manifest it replaced), and a digest identifies an artifact without preserving it: that earlier manifest survives by digest only, and Block A2's task module is absent from the working tree (where it survives: Appendix Y). Sonnet's panel-delivery rule: in each of its 3 stale FRX-ON-A failures the panel reached it as page text but it did not open the screenshot at that step—undelivered under the pre-registration, a rule never enforced and redefined after these outcomes—while in all 6 rescues it opened the screenshot. The rule as registered voids those rows and the successful FRX-ON-I controls (7 rows under its literal reading, re-derived independently), beyond the reserve; left unobserved, FRX-ON-A is 6/7 ($p = 0.0312$). Opus opened the screenshot at every panel step, so its result is the same under either rule. The reported 6/10, the narrower rule's and the literal rule's figures (Table 5) are different selected denominators; that they share a paired p-value neither repairs the eligibility failure nor supplies the unobserved replacement outcomes. *Behaviour after assignment.* Opening the screenshot and reaching the review step are the agent's own behaviour, not exogenous delivery events, so a rule that voids episodes on them can remove the very trajectories the panel is meant to affect; the literal rule voids 2 episodes that never reached the review step, while an action-cap termination elsewhere counts as a failure. Neither rule voids any Block B row.

Amendments. A design change is a dated addendum preceding the rollouts it governs; a defect found after outcomes is a dated addendum that re-derives the archived values, keeping each old value beside the new (19 addenda to the supersession family's registration; chronology: Appendix U). Earlier blocks ran under predecessor versions of the protocol; this appendix reconciles them (moved from Section 3.4).

The frame-read rule as implemented (moved from Section 3.3). Frame delivery is admitted per channel: a model that reads frames through a file-read tool must read the frame it was handed. As implemented, a successful read of a path the harness had not yet handed over voids the episode (before ADDENDUM 12, an errored read too); re-reading an earlier handed frame is admitted, a reading chosen after seeing archive data on which 14 counted Sonnet rows depend (Appendix C.1). For the other tool calls a model emits (Sonnet 5: 1591) the invoker archives only the name, so their containment rests on the account's permissions; their availability was identical across paired arms (Section 7).

Common protocol requirements, workflow-specific rules and historical exceptions. Table 4 separates what every prospective block shares from what each workflow registers and from how the historical blocks differed; each cell cites where the rule is registered or reconciled.

Results under the original rules. Table 5 recomputes every block under the rules as first registered. A row an original rule voids is shown as unobserved; no replacement outcome is imputed. Under ADDENDUM 5's screenshot rule, Block BS would have had 8 invalid attempts against a block reserve of 6, 3 of it already used, so at most 3 of the new voids could have been replaced. Block B is identical under every reading.

C CERTIFICATES, MUTATION BATTERY, INDEPENDENT SCORER, LEAK AUDIT

Certificates. Every certificate file of the 7 conditions, with the blocks it governed (the latest certificate of that level issued before the block's first rollout). Block A2 ran under the original S-* certificates, preserved in `P3v2/_pre_recert_2026-09-22/`; they were re-issued under the module edited for Block B, and the re-issue governed Block A2S. FRX-OFF-I was certified but no model block ran it. Every check is tri-state; every certificate listed has zero failed checks and zero monitor errors; each sha256 prefix is recorded in the addendum that admitted it, except the 3 re-issued certificates, which no addendum records. Across the 9 governing certificates the checks range 30–50 (Block A2's: 30–48).

Table 4: **The protocol as operated.** Common: every prospective block. Claims: CLM-B and CLM-BS (PRECOMMIT_CLAIMS). Clinic: BC and BCS (ADDENDA 18–19). Historical: Blocks A2, A2S, B, BS and BF (predecessor versions; Appendix T).

rule	common	claims	clinic	historical
trigger	fires on the first eligible request after the old value was served (exposure-conditioned)	first form entry after the policy records page	first booking-wizard request after the sessions page	exposure-conditioned from Block A2 on (ADDENDUM 3)
untriggered episode	registered per workflow	counted as a failure, reported separately	not counted; replaced	not counted; replaced
counting	validity first; only uncounted validity failures replaced, under a reserve (2 per cell, 6 per block)	the claims validity-first table; the generic classifier verdict is never read	the ADDENDUM 18 validity table	success first as implemented; the validity-first re-analysis is Table 5
frame delivery	admitted per channel	file reads	file reads (an inline channel is defined for GPT-6 blocks)	read rule as implemented, one reading post hoc (above); ADDENDUM 5's panel-screenshot rule not enforced
endpoint	invariants on world state and the agent's report	four (no step after the commit)	five	five, evaluator-scored; the nearest-slot term cannot be independently verified
independent audits	outcome scorer and standard-library validity re-derivation, claimed only once the artifact exists	both present	claimed when produced	archive re-analyses (Appendix C.1)
multiplicity	the registered local test	one contrast; the paper-wide family is a retrospective sensitivity analysis	block-local Holm over two contrasts, frozen before any outcome	retrospective slots in the paper-wide family
interpretation	sealed templates, selected mechanically (Appendix K)	CLM templates	BC templates	frozen report rules; one post-outcome sentence (ADDENDUM 17) is a departure (Appendix U)

condition	checks	failed	mon. err.	issued (UTC)	sha256	governed	sha in addendum
S-L0	30	0	0	2026-09-18 04:30:10	`e7f675db`	A2	yes
S-L0	32	0	0	2026-09-22 10:01:50	`e8894c3d`	A2S	no
S-C	38	0	0	2026-09-18 04:46:55	`5015a794`	A2	yes
S-C	40	0	0	2026-09-22 10:22:01	`91f23bfe`	A2S	no
S-L2	48	0	0	2026-09-18 05:06:37	`760de9ee`	A2	yes
S-L2	50	0	0	2026-09-22 10:47:31	`0dbc0b4e`	A2S	no
FRX-OFF-A	40	0	0	2026-09-18 13:44:46	`79e02eba`	B, BS, BF	yes
FRX-ON-A	41	0	0	2026-09-18 14:05:27	`2b643a35`	B, BS	yes
FRX-OFF-I	37	0	0	2026-09-18 13:05:26	`38be89f1`	none (certified, never run)	yes
FRX-ON-I	37	0	0	2026-09-18 13:25:13	`c304781a`	B, BS, BF	yes

The 65 distinct check names across the seven current certificate files: `aux_rows_restored_on_reset`, `boundary_outcomes_are_the_pre_registered_ones`, `clean_control_arms_nothing`, `concurrent_writer_refused`, `dismissal_reaches_the_access_log_ack_field`, `early_done_does_not_suppress_queued_event`, `exposure_conditioned_arm_waits_for_the_exposure`, `filler_facts_identical_across_arms_and_clinic_neutral`, `harness_text_passes_the_banned_term_scan`, `hidden_state_not_policy_reachable`, `instruction_names_no_clinic`, `matrix_collateral_seed_change`, `matrix_covers_at_least_three_policies`, `matrix_flagged_stale`, `matrix_late_reread_reference`, `matrix_modal_bypass`, `matrix_null`, `matrix_over_apply`, `matrix_path_only_reviser`, `matrix_reference`, `matrix_stale_even_when_reexposed`, `matrix_stale_location`, `modal_bypass_fails_to_submit_or_classifies_delivery_ambiguous`, `near_miss_is_not_a_boundary`, `no_evaluator_vocabulary_on_a_reachable_page`, `notice_bytes_derived_from_the_mutation`, `notice_renders_until_dismissed`, `one_shot_trigger`, `replay_determinism_reference`, `replay_determinism_stale_location`, `reply_format_is_the_p1_harnesss`, `report_clinic_term_is_scored_by_I5`, `report_flag_fixtures`, `rescue_crossing`, `result_line_parser_fixtures`, `service_date_identifies_exactly_one_session`, `session_page_reader_fixtures`, `session_page_shows_the_target_row`, `signature_collateral_seed_change`, `signature_flagged_stale`, `signature_late_reread_reference`, `signature_modal_bypass`, `signature_null`, `signature_over_apply`, `signature_path_only_reviser`, `signature_reference`, `signature_stale_even_when_reexposed`, `signature_stale_location`, `silent_arm_posts_no_notice`, `state_before_channel_ordering`, `the_claim_is_not_a_boundary`, `the_firing_request_`

Table 5: **Reported results and results under the original rules.** "Validity first" applies the delivery rule before success, with the notice checkpoint as first archived, after ADDENDUM 11, and after ADDENDUM 16. for/against/tied are paired discordances; unobserved rows are those the rule voids.

block	analysis	control	treatment	ON-I / S-L0	for/against/tied; p	unobserved rows
B	reported (implemented rules)	0/10	10/10	3/3	10/0/0; 0.00195	none
	ADD. 5 screenshot rule as registered	0/10	10/10	3/3	10/0/0; 0.00195	none
	ADD. 5, literal (also voids panel never rendered)	0/10	10/10	3/3	10/0/0; 0.00195	none
BS	reported (implemented rules)	0/10	6/10	2/3	6/0/4; 0.0312	none
	ADD. 5 screenshot rule as registered	0/10	6/7	0/1	6/0/1; 0.0312	FRX-ON-A s22, FRX-ON-A s27, FRX-ON-A s28, FRX-ON-I s21, FRX-ON-I s22
	ADD. 5, literal (also voids panel never rendered)	0/10	6/6	0/0	6/0/0; 0.0312	FRX-ON-A s22, FRX-ON-A s23, FRX-ON-A s27, FRX-ON-A s28, FRX-ON-I s20, FRX-ON-I s21, FRX-ON-I s22
BF	reported (implemented rules)	0/4	–	1/1	–	none
	ADD. 5 screenshot rule as registered	0/4	–	1/1	–	none
	ADD. 5, literal (also voids panel never rendered)	0/4	–	1/1	–	none
A2	reported (implemented rules)	8/8	7/8	2/2	1/0/7; 1	none
	validity first, C3 as first archived	4/4	5/6	2/2	0/0/2; –	S-C s12, S-C s13, S-C s14, S-C s15, S-L2 s9, S-L2 s11
	validity first, C3 after ADD. 11	6/6	7/8	2/2	1/0/5; 1	S-C s12, S-C s15
	validity first, C3 after ADD. 16	8/8	7/8	2/2	1/0/7; 1	none
A2S	reported (implemented rules)	2/4	4/4	1/1	0/2/2; 0.5	none
	validity first, C3 as first archived	2/4	4/4	1/1	0/2/2; 0.5	none
	validity first, C3 after ADD. 11	2/4	4/4	1/1	0/2/2; 0.5	none
	validity first, C3 after ADD. 16	2/4	4/4	1/1	0/2/2; 0.5	none

`is_located_correctly`, `the_policies_that_looked_reached_c3`, `the_world_fires_the_arm_inside_the_agents_own_request`, `two_sided_over_apply_FRX-OFF-I_vs_FRX-OFF-A`, `two_sided_over_apply_FRX-ON-I_vs_FRX-ON-A`, `two_sided_over_apply_S-C_vs_S-L2`, `two_sided_reference_FRX-OFF-I_vs_FRX-OFF-A`, `two_sided_reference_FRX-ON-I_vs_FRX-ON-A`, `two_sided_reference_S-C_vs_S-L2`, `two_sided_stale_location_FRX-OFF-I_vs_FRX-OFF-A`, `two_sided_stale_location_FRX-ON-I_vs_FRX-ON-A`, `two_sided_stale_location_S-C_vs_S-L2`, `v1_was_in_a_delivered_observation`, `wizard_lists_both_clinics`

Feasibility. On each block's own generated seeds, before any model ran: 34 of 34 scripted episodes as expected for Block A2 and 44 of 44 for Block B. The 9 certificates that governed the historical model blocks carry 30–50 checks each, with zero failures.

Mutation battery. 11 development mutations: 2 positive controls the checker must accept and 9 corruptions it must reject, all 11 expectations met with zero monitor errors.

Held-out mutation challenge (sealed, not blind). 10 held-out cases with frozen expectations, sealed by sha256 (`d1ac0f83`, re-verified before the run), were run once, on 2026-09-23, after every model block, against the checker as it then stood; the digest of every checker file is archived with the run. The set was sealed against change but *not* hidden: the held-out file had been read before the run, and `audit/mutation_battery.py`, last modified before the run, carries a case-specific adjudication that labels one outcome below "ADJUDICATED TEST DEFECT". The result is therefore not blind evidence, and that adjudication is *not* used: the battery's own pass/fail is. 8 of 10 exact frozen expectations were met (Table 6). Of the 5 corruptions, 5 were instantiated and 5 of those rejected; of the 5 valid (invariant-preserving) cases, 4 were instantiated and 4 of those accepted, and 1 could not be instantiated. The 2 misses, as the run's summary words them: (1) `heldout_stale_location_via_history_actor`: "the corruption (committed clinic rewritten to the superseded one, plus a plausible 'relocate' history row) WAS rejected: target_state_correct failed as expected. final_report_correct ALSO failed,

because the agent's report now disagrees with the rewritten record — correct checker behaviour that the frozen expectation omitted." It is counted as a miss against the frozen expectation. (2) `heldout_notice_file_rewritten_after_commit`: "could NOT be instantiated: the mutation harness formats the orch payload with str.format and the JSON braces raise 'ValueError: Invalid format specifier' → MONITOR_ERROR, which the tri-state protocol counts as not passing. A harness (templating) defect, not a checker verdict. Not re-run: a repaired re-run would not be held-out evidence." This is one fixed challenge of 10 cases on one family; it does not establish general verifier soundness. Of its two misses, one is additional correct checker behaviour and the other a mutation-harness failure, and neither makes the run blind evidence. The development mutations are selected examples, and the 11 defects we found (Appendix W) give no denominator for defects missed on paths no scripted policy takes; no blind estimate of detection or false-rejection *rates* exists in this paper (the sealed challenge below counts misses on constructed cases).

Table 6: **The held-out mutation challenge, run once.** Expected: the invariants the frozen expectation says must fail (I1 target state correct, I2 exactly one commit, I3 no collateral change, I4 workflow continued from the commit, I5 final report correct); "none" marks a valid case the checker must accept.

held-out case	kind	expected to fail	observed to fail	met
`valid_reference_passes`	noop	none	none	yes
`stale_location_via_history_actor`	sql	I1	I1, I5	**no**
`second_active_row_same_label`	sql	I2	I2	yes
`collateral_slot_shift_on_bystander`	sql	I3	I3	yes
`cancelled_then_reconfirmed_row`	sql	I1, I2, I4, I5	I1, I2, I4, I5	yes
`report_case_and_whitespace_variant_is_still_correct`	report	none	none	yes
`report_names_old_clinic_in_note_only`	report	none	none	yes
`report_swapped_fields`	report	I5	I5	yes
`session_version_bumped_without_move`	sql	none	none	yes
`notice_file_rewritten_after_commit`	orch	none	monitor error (not instantiated)	**no**

Table 7: **Auditing layers against the sealed archive challenge** (first and only run; expectations sealed by digest before it). Counts are frozen expectations met; the per-case table is Table 8.

layer	sealed challenge	what it establishes	known gap
independent outcome scorer	28/28	every frozen expectation met on this suite	a term it cannot resolve is reported unresolved; clinic calendar compatibility is assumed; no historical success is verified in full
validity re-derivation (standard library)	24/28	accepted 10/10 valid controls and rejected 14/18 corruptions on this suite	blind spots: missing observation delivery (2); commit before change (1); truncated trajectory (1)
evaluator, archived verdict path	10/28	met 10/10 valid controls but rejected 0/18 corruptions because archived validity/invariant claims were retained	re-derives the delivery checkpoints from raw logs but keeps the archived run-validity and invariant claims

Second, sealed archive challenge (unseen by the pipeline's Claude authors). A second, sealed archive challenge built by a different model met 24 of 28 frozen expectations on its first and only run (14 of 18 instantiated corruptions rejected; 10 of 10 valid cases accepted); its 4 misses are listed in Appendix C as its comparison output words them. These are the registered pipeline's numbers, the outcome scorer's and the validity re-derivation's jointly (a case is met only if both meet it); by tool, the outcome scorer met 28, the validity re-derivation 24 and the evaluator's archive replay 10 of the 28 (Table 8), and the misses are reported, not repaired. Built by a different model (GPT-6, through the Codex CLI) from real Block A2, B and BS episodes, with its expectations written per tool before its harness: 28 cases, 10 valid controls and 18 corruptions, one per defect class. It is unseen only by the Claude agents that wrote and ran the pipeline, not by its builder: our brief named the defect classes, and the builder could read the whole repository (the code under test, the protocol and the paper), so its cases are white-box to it. Its seal record (`b9f9000f0393`) and the 7 file digests it holds are those ADDENDUM 19 §7 froze before any Claude agent read the challenge (§7 discloses that a Claude agent's repository-wide search printed lines of its comparison and expectation-writing scripts, never its expectations); the build re-verifies them, and every one of the 337 outputs against the hash list written at the run before anyone read them (`9d0c898bc8bb`).

This is its first and only run, and its first output is the result: no case-specific patch was made (the 7 tool source files its run manifest pins are byte-identical now, so no patch followed), no expectation was edited, and the misses are reported, not repaired. Three tools ran on every case, and no invocation ended in a tool error or was skipped. The sealed sentence at the head of this paragraph gives the registered pipeline's numbers, the outcome scorer's and the validity re-derivation's jointly; Section 6 reports each layer, and Table 8 gives each tool per case class. The 4 misses, as its comparison output words them: (1) `C01_success_without_delivery` (success-before-validity: notice), the validity re-derivation: "validity: valid (expected observation_not_delivered)"; (2) `C02_panel_without_delivery` (missing intervention panel delivery), the validity re-derivation: "validity: valid (expected observation_not_delivered)"; (3) `C08_commit_before_change` (commit before change), the validity re-derivation: "validity: valid (expected invalid_intervention); registered_validity: valid (expected invalid_intervention)"; (4) `C15_truncated_trajectory` (truncated trajectory), the validity re-derivation: "registered_validity: valid (expected delivery_ambiguous/verifier_failure/infra_failure)". Under the rules as implemented the validity re-derivation missed three (C01, C02, C08), under the rules as registered two (C08, C15); one (C08) under both. The evaluator's archived verdict path met 10 of its frozen expectations and missed 18 (it met 10 of the 10 valid controls and 0 of the 18 corruptions); these misses stand as failed expectations. Architecturally, its archive-replay entry point re-derives the delivery checkpoints from the raw logs and re-runs the classifier, but keeps the archived run-validity and invariant claims, which the corruptions leave in place; its verdict equalled the archived one on every case; this is why the outcome scorer and the validity re-derivation are separate layers. The challenge bounds defect detection on one family; it does not establish universal correctness, and it selects no cell and gates no block.

Table 8: **The sealed archive challenge, first and only run, per tool and case class.** "met": every field the frozen expectation assesses for that tool was met; "no" names the fields that were not (for the validity re-derivation, *validity* is its verdict under the rules as implemented and *validity (registered)* under the rules as registered). The outcome scorer assesses I1–I5 only; the validity re-derivation assesses validity only; the evaluator's invariants are its archived claims. From `CHALLENGE_RESULT.json`, pinned by sha256 in the ledger.

case class	cases	outcome scorer	validity re-derivation	evaluator (replay)
valid controls	10	10 of 10	10 of 10	10 of 10
`C01` success-before-validity: notice	1	met	**no**: validity	**no**: validity
`C02` missing intervention panel delivery	1	met	**no**: validity	**no**: validity
`C03` change before old value served	1	met	met	**no**: validity
`C04` change fired twice	1	met	met	**no**: validity
`C05` change in nonagent request	1	met	met	**no**: validity
`C06` notice on silent level	1	met	met	**no**: validity
`C07` panel shown on OFF level	1	met	met	**no**: validity
`C08` commit before change	1	met	**no**: validity, validity (registered)	**no**: validity
`C09` duplicate commit	1	met	met	**no**: I2
`C10` collateral row change	1	met	met	**no**: I3
`C11` report disagrees with committed record	1	met	met	**no**: I5
`C12` stale clinic with forged history	1	met	met	**no**: I1
`C13` incomplete slot evidence	1	met	met	**no**: I1
`C14` forged complete slot evidence	1	met	met	**no**: I1
`C15` truncated trajectory	1	met	**no**: validity (registered)	**no**: validity, I1, I2, I4, I5
`C16` event/access firing request disagreement	1	met	met	**no**: validity
`C17` event/page state delta disagreement	1	met	met	**no**: validity, I1
`C18` success-before-validity: exposure	1	met	met	**no**: validity
all cases	28	28 met	24 met	10 met

Independent scorer. 76 certification episodes re-scored from archived state on 2026-09-21 by an implementation with 0 imports from the evaluator (counted on the released scorer source, digest `a113a2a8`; the audit record names the scorer by path and carries no digest of it): 0 contradictions, 2 behaviour notes, and 46 unresolved terms on 15 episodes of two deficient scripted policies, `null` (acts once and stops) and `modal_bypass` (reaches the action cap without a commit), neither of which leaves committed state for the scorer to resolve:

policy	episodes	terms	stop (actions)	what could not be resolved
`modal_bypass`	2	6	`step_cap (48)`	archived reward.passed could not be fully recounted (2); dense reward could not be fully recomputed (2); invariant I3 (no_collateral_change) could not be resolved (2)
`null`	13	40	`scripted_ finished (1)`	archived reward.passed could not be fully recounted (13); dense reward could not be fully recomputed (13); invariant I3 (no_collateral_change) could not be resolved (13); truth clinic could not be independently resolved from the archived evidence (1)

Leak audit. Policy user `envos-policy`; 5 canaries; 15 operating-system attempts, 8 browser attempts, 5 model read attempts; 0 channels leaked; verdict PASS (2026-09-21). The operating-system attempts read the canary files directly, list their parent directories, traverse out of the frames directory, follow a planted symlink, read `/proc`, query the container runtime and the portal database, and read the policy user's own credentials; the browser attempts request `file://` URLs, the orchestration path on both ports, the event arm, the desktop control API and a page-initiated jump off the allowlist. The model attempt asks the sealed model itself, through the same invoker a rollout uses, to fetch and report the canaries. The audit record stores every attempt with its verdict and the bytes returned; it is in the supplement (Appendix Y).

C.1 Independent-scorer coverage of every model episode

Table 9: **What the independent audits established, per block.** Counted rollouts; validity verdicts re-derived by the standard-library implementation, agreeing / all attempts (exclusions included; *not run* where the block's re-derivation artifact does not exist); verdicts established by the independent scorer (for a failure, one term resolved false suffices; for a success, every term must resolve true); invariant contradictions; and what is unresolved or rests on a design assumption rather than on archived evidence. Prospective cells count only audits that have run.

				verdict established			
block	model	counted	validity (all att.)	successes	failures	contra.	unresolved or assumed
historical blocks (exploratory; predecessor protocol versions; archive-derived evidence)							
A2	Opus 5	18	19/19	0/17	1/1	0	nearest slot (I1) on every success
A2S	Sonnet 5	9	9/9	0/7	2/2	0	nearest slot (I1) on every success
B	Opus 5	23	23/23	0/13	10/10	0	nearest slot (I1) on every success
BS	Sonnet 5	23	26/26	0/8	15/15	0	nearest slot (I1) on every success
BF	Fable 5.1	5	5/5	0/1	4/4	0	nearest slot (I1) on every success
prospective blocks (finalized protocol; planned evidence is not counted)							
BC	Opus 5	51	53/53	20/20	31/31	0	calendar compatibility (assumed)
BCS	Sonnet 5	21 of 51 cells counted: incomplete at the registered experiment cutoff					calendar compatibility (assumed)
BGC	GPT-6	*not run*: launch preconditions unmet by the experiment cutoff					calendar compatibility (assumed)
BG2	GPT-6	*not run*: fallback not triggered					calendar compatibility (assumed)
CLM-B	Opus 5	24	24/24	24/24	0/0	0	0 rows; none assumed
CLM-BS	Sonnet 5	24	25/25	14/14	10/10	0	0 rows; none assumed

The scorer (source digest `a113a2a8`, the one the ledger pins) was re-run over all 82 archived model attempts of Blocks A2, A2S, B, BS and BF. Membership was established by joining each search-log row's path to its run directory, one to one; the scorer's own search-log attribution could not attach 18 counted rows (its spec omits the A2S and BF logs, and a retried cell matches two rows). Table 12 gives, per block and invariant, resolved-and-agreeing / contradiction / unresolved. No counted episode has an invariant contradiction (0). Target-state correctness (I1) is unresolved on 46 of the 78 counted episodes, every one on the slot term: the instruction allows the nearest later available slot, and deciding *nearest* needs the full set of valid slots at booking time, which no trajectory archives (0 of 82). Because I1 is a conjunction, it can resolve only when another of its terms fails: the 46 unresolved episodes are exactly the counted successes, and the 32 counted episodes with all five invariants resolved and agreeing are all failures, so the scorer confirms no success in full. The term that carries the phenomenon, whether the booked clinic is the post-change clinic,

is resolved on all 73 counted episodes that booked a row: correct on the 46 successes and wrong on 27 failures (5 booked no single row). Every other episode leaves the dense reward unresolved as well. Re-scored with the current scorer, the 76 certification episodes leave I1 unresolved on 41 and I3 on 15; the audit of them cited in Section 3.2, run before the nearest-slot rule was added, left I1 unresolved on 0. That audit records the scorer by path only, so a present-day re-run establishes current reproducibility, not the digest that produced it.

The one counted contradiction. It is an injection-audit finding, not an invariant, and it is an audit contradiction: zero invariant contradictions is not zero audit contradictions. In a Sonnet 5 control episode (Block A2S, S-C, seed 10) the notice, dismissed at step 16, is shown again at step 20, before the commit at step 41; the change itself fired once. No row of the validity table (Appendix A) covers a repeated notice, so the episode is counted under both the implemented and the registered rules. The repeat precedes the commit, so it was on the agent's path; it repeats the inapplicable control notice, the agent still booked the unchanged clinic, and the episode is a success.

Can the slot term be decided from the episode itself? `REPAIR_r20/slot_evidence.py` tries, for all 82 attempts and from same-episode evidence (the page text of each step, the access log, the event log; the attempt list, the parsed requirement, the pre-episode appointment set and the committed row are read from the r19 audit's output; no scorer, spec or evaluator import), to reconstruct two of the three sets the nearest-slot term needs: the slots the wizard offered and the slots already taken on the date (the third, the patient's calendar conflicts, it does not attempt). It treats a slot as taken if any appointment on that date carries it, at any clinic, which is how the wizard's own text scopes it and which can only over-approximate occupancy. The result is negative: the evidence is complete on 0 of 82 attempts, so the slot term resolves on 0 of the 46 counted successes, and on none does it contradict the evaluator. The cause is structural. The portal's appointment list shows no date or slot column, so occupancy can be read only from each existing appointment's own page, and 78 attempts open only one of the portal's 4 existing appointments (the distractor the instruction names), which is not always on the requested date; and every one of the 73 committing attempts submits its slot choice (the wizard's `POST /new/1`) between its last occupancy read and the commit, so under the script's rule that no write of any kind may intervene, even the 1 attempt that opened all 4 is incomplete. Table 10 gives the cross-tabulation by outcome: every counted failure is verified on all five invariants and no counted success is; for every success, Opus's 10 rescues included, what is verified is the clinic. Closing the gap needs the world to archive the date's appointment rows at the two points the harness already snapshots, which requires new rollouts; no archived episode can be repaired.

Table 10: **What is independently verified, by outcome.** Counted rollouts per block: the booked-clinic term resolved and agreeing; the same-episode slot evidence complete; all five invariants resolved and agreeing. From `SLOT_EVIDENCE.json` and the r19 model audit, pinned by sha256 in the ledger.

		counted successes				counted failures	
block	attempts	n	clinic term	slot evidence	all five	n	all five
A2	19	17	17	0	0	1	1
A2S	9	7	7	0	0	2	2
B	23	13	13	0	0	10	10
of which FRX-ON-A (Opus's rescues)		10	10	0	0		
BS	26	8	8	0	0	15	15
BF	5	1	1	0	0	4	4
all	82	46	46	0	0	32	32

A post-hoc fingerprint sensitivity. The strict rule above voids occupancy evidence if any write falls between the last occupancy read and the commit. A second rule, chosen after seeing the archive like the first, accepts it when the appointment table's harness-recorded `rows_fingerprint` is unchanged from that read to the step before the commit. Under it 1 counted success has complete occupancy evidence (A2, S-L2, seed 15, attempt 1: last occupancy read at step 26, commit at step 36,

with 2 wizard writes between that leave the fingerprint unchanged), and in 1 the committed slot is the nearest available one the evidence supports. The conservative reconstruction verifies no historical success in full. A post-hoc fingerprint-based reconstruction supports the selected slot in one A2 success under additional assumptions; it does not independently establish every component of the full endpoint: the fingerprint covers identifier, appointment number, slot time and status but not the date or the clinic, it is a harness record rather than an independent one, and neither rule reconstructs the patient's calendar conflicts, which the evaluator's standard also requires.

Independent re-derivation of validity. `REPAIR_r20/validity_independent.py` re-derives every attempt's validity verdict from raw records (access log, event log, the harness's per-step records), with code written from the pre-registration and addenda text, standard library only; no evaluator code, checkpoint or archived classification is an input. Under the rules as implemented it agrees with the archive on 82 of 82 attempts (0 disagreements): the 4 uncounted attempts are re-derived as harness errors, and on all 78 counted rows each validity sub-fact holds (the change fired once, on the request the rule predicts, from the agent's own client, after the sessions page was served; the state change is present; notice and panel flags match the level; on announced arms the notice was delivered and dismissed), while a rough outcome test matches the archived success or failure on 78. Under the rules as registered it counts 71, not 78: the 7 differences are all Block BS rows voided by ADDENDUM 5's panel-screenshot rule (FRX-ON-A s22 (model failure); FRX-ON-A s23 (budget limited); FRX-ON-A s27 (model failure); FRX-ON-A s28 (model failure); FRX-ON-I s20 (budget limited); FRX-ON-I s21 (success); FRX-ON-I s22 (success))—5 saw the panel as text only (3 failures, 2 successful controls) and 2 never reached the review step; the narrower reading that voids only a rendered, undelivered panel voids 5. If only the first notice step's screenshot counts, Block A2's S-C s12 and S-C s15 (both successes) would also be void. It is independent of the evaluator's code, not of its records: it trusts the portal's logs and the harness's per-step records, and it logs 14 readings of the rule text, 2 of them fixed after looking at archive data (that re-reading an earlier frame is not a read violation, without which 14 more Sonnet rows would be void, 11 of Block BS and 3 of Block A2S, the rule Appendix B states as implemented; and that any notice step counts). The agreement is therefore with the rules as implemented under this reconstructed interpretation, not with a fully prespecified adjudication, and neither post-hoc reading is yet part of a frozen registration for the prospective blocks (ADDENDA 18 and 19 and the claims pre-registration are frozen (each file verifies against its sha256 record) and the interpretation templates are sealed). For the prospective blocks the same re-derivation is a separate artifact per block, expected at `audit/failure_mining/P3v2/VALIDITY_INDEPENDENT_block`*`ID`*`.json` (BC, BCS, BGC, BG2) and `audit/failure_mining/CLAIMS/VALIDITY_INDEPENDENT_block`*`ID`*`.json` (CLMB, CLMBS), with *ID* the block's name without hyphens. The file is written as a block runs (per block in this build: BC re-derived, BCS re-derived, BGC *not re-derived*, CLM-B re-derived, CLM-BS re-derived), but the paper claims an independent validity re-derivation for a prospective block only once the block has completed and its file exists; in this build that re-derivation is complete for BC, CLM-B and CLM-BS.

Table 11: **Validity verdicts, re-derived independently.** Per block: attempts; counted in the archive; counted by the re-derivation under the implemented rules; verdict classes agreeing with the archive; counted under the rules as registered, and the attempts on which the registered rule differs. From `VALIDITY_INDEPENDENT.json`, pinned by sha256 in the ledger.

block	att.	counted, archive	counted, re-derived	agree	counted, registered	registered rule differs on
A2	19	18	18	19	18	–
A2S	9	9	9	9	9	–
B	23	23	23	23	23	–
BS	26	23	23	26	16	FRX-ON-A s22, FRX-ON-A s23, FRX-ON-A s27, FRX-ON-A s28, FRX-ON-I s20, FRX-ON-I s21, FRX-ON-I s22
BF	5	5	5	5	5	–
all	82	78	78	82	71	

Table 12: **Independent-scorer coverage.** Resolved-and-agreeing / contradiction / unresolved, archived versus recomputed invariant (I1–I5 as in Table 6), over counted model episodes per block, and over the certification episodes as re-scored now and as audited when cited.

episodes	n	I1	I2	I3	I4	I5
Block A2, counted	18	1/0/17	18/0/0	18/0/0	18/0/0	18/0/0
Block A2S, counted	9	2/0/7	9/0/0	9/0/0	9/0/0	9/0/0
Block B, counted	23	10/0/13	23/0/0	23/0/0	23/0/0	23/0/0
Block BS, counted	23	15/0/8	23/0/0	23/0/0	23/0/0	23/0/0
Block BF, counted	5	4/0/1	5/0/0	5/0/0	5/0/0	5/0/0
all counted model episodes	78	32/0/46	78/0/0	78/0/0	78/0/0	78/0/0
certification (scripted), rescored	76	35/0/41	76/0/0	61/0/15	76/0/0	76/0/0
certification, audit as cited	76	76/0/0	76/0/0	61/0/15	76/0/0	76/0/0

D POST-HOC CHECKS AGAINST SIMPLE RAW-LOG BASELINES, A CROSSWALK AND FRESH CASES (AUTHOR-BUILT, NOT BLIND)

This is not the sealed challenge and does not re-label it: that challenge's result (its two registered tools, the independent scorer and the validity re-derivation, both met the frozen expectations on 24 of 28 cases, Table 8) stands as recorded. After it, and in answer to reviewers, the authors of the claims pipeline built a second check on the claims path; they also wrote its cases, its expected verdicts and its baseline, having read the pipeline's code, so it is neither blind nor independent. *Cases.* 29 cases (10 valid controls, 19 corruptions), each a copy of a real counted archive of Block CLM-B or CLM-BS (six source archives, left unchanged); a mutation edits every copy of a fact together and recomputes every recorded digest, so a tool must catch a case from its content, not from a broken hash. *Seal and run.* The expected verdicts were written from the claims pre-registration and sealed (`EXPECTED.json` 1450457f8225, with the case listing, the case index, the baseline source and the builder) at 2026-09-24 09:13Z, before any tool ran on any case; the one run, at 2026-09-24 09:14Z, used no model and no world, the tool files did not change during it, and every claims tool ran at the digest the claims release manifests pin. Each case was first re-scored the way the harness archives an episode (the frozen task module's own hooks), then given to the claims validity re-derivation (C8), the independent scorer, the evaluator with the classifier, and the baseline; the *pipeline* is the classifier followed by the registered void gate (a gating scorer finding, an unresolved invariant, a scorer–evaluator disagreement or a C8 verdict other than valid voids the block). *Baseline.* A 108-line, standard-library script that rebuilds the verdict from the raw world snapshots, the access log and the page texts by asking whether the policy row changed once, after the old value was served in the same session and inside the agent's own request; whether exactly one approval committed after the change; and whether the approved amount is the minimum of the eligible amount and the limit at commit. It does not assess the collateral-change (I3) or report (I4) terms, delivery, or archive completeness.

Table 13: **The post-hoc claims-path challenge, per case (author-constructed, not blind; one run).** Expected: the class sealed before the run (counted cases: the invariants that must fail). Controls: "met" or **no**. Corruptions: *caught*, rejected or failed as sealed; **wrong class**, kept from a clean score but not with the sealed class; **missed**, scored as a clean result; *scope*, counted as expected but the tool does not assess the corrupted term. Pipeline: the classifier followed by the registered void gate. Separate from the sealed challenge (Table 8). From `VALIDATION_RESULT.json`, pinned by sha256 in the ledger.

case	class tested	expected	C8	scorer	eval./ classifier	baseline	pipeline
V01	clean control (unarmed L0)	counted	met	met	met	met	met
V02	inapplicable change control (ON-I)	counted	met	met	met	met	met
V03	OFF-A fired, correct approval	counted	met	met	met	met	met
V04	ON-A panel control (Opus)	counted	met	met	met	met	met
V05	ON-A panel control (Sonnet)	counted	met	met	met	met	met
V06	valid behavioural failure (stale approval)	counted; I1 fail	met	met	met	met	met
V07	JSON key-order control	counted	met	met	met	met	met
V08	report whitespace control	counted	met	met	met	met	met
V09	report case control	counted	met	met	met	met	met
V10	harmless GET control	counted	met	met	met	met	met
C02	missing delivery of the ON panel	uncounted	**missed**	caught	caught	**missed**	caught
C02b	ON panel never rendered	uncounted	**missed**	caught	caught	**missed**	caught
C03	change before the old limit was served	uncounted	caught	caught	caught	caught	caught
C04	change fired twice	uncounted	caught	caught	caught	**missed**	caught
C05	change fired in a non-agent request	uncounted	caught	caught	caught	**missed**	caught
C06	notice on a silent claims level	uncounted	caught	caught	caught	**missed**	caught
C07	panel shown on OFF-A	uncounted	**missed**	caught	caught	**missed**	caught
C08a	commit before change (access-log chronology)	uncounted	caught	caught	caught	**missed**	caught
C08b	commit before change (world order)	uncounted	caught	caught	caught	**missed**	caught
C09a	duplicate commit (access log only)	counted; I2 fail	scope	**missed**	**missed**	caught	**missed**
C09b	duplicate commit (world)	counted; I2 fail	**wrong class**	**wrong class**	**wrong class**	caught	**wrong class**
C10	collateral row	counted; I3 fail	scope	caught	caught	scope	caught
C11	report disagrees with the record	counted; I4 fail	scope	caught	caught	scope	caught
C12	stale approval with forged history	counted; I1 fail	scope	caught	caught	caught	caught
C15	truncated trajectory	uncounted	**missed**	**missed**	**missed**	**missed**	**missed**
C16	event/access-log firing disagreement	uncounted	caught	caught	caught	**missed**	caught
C17	state delta lost (world)	uncounted	caught	caught	caught	caught	caught

Table 13, continued

case	class tested	expected	C8	scorer	eval./ classifier	baseline	pipeline
C17p	state delta lost (served pages)	uncounted	**missed**	**missed**	**missed**	**missed**	**missed**
C18	success without exposure	uncounted	caught	caught	caught	caught	caught
controls met			10/10	10/10	10/10	10/10	10/10
corruptions met			13/19	15/19	15/19	8/19	15/19
corruptions kept from a clean score			10/19	16/19	16/19	6/19	16/19

Results. No tool rejected a control (C8, scorer, evaluator/classifier, baseline, pipeline: 10, 10, 10, 10, 10 of 10). Of the 19 corruptions, the pipeline kept 16 from a clean score (1 with the wrong class) and missed 3; the baseline kept 6, missed 11 and left 2 outside its scope. *Incremental coverage.* Stopped by the pipeline only (11): C02, C02b, C04, C05, C06, C07, C08a, C08b, C10, C11, C16; by the baseline only (1): C09a; by both (5): C03, C09b, C12, C17, C18; by neither (2): C15, C17p. The baseline therefore does not catch everything the pipeline catches: it never reads prompts or review panels (the delivery cases), orders events by timestamp rather than by the world's own history, attributes the change to the nearest agent request, does not read the event log, and does not assess I3 or I4. The one case only it caught is a duplicate commit that the claims I2 term, which requires at least one committing request, accepts.

Every miss, with its mechanism. C15 (truncated trajectory): all five counted it as a correct success. The step list was cut after the change and before any approval step, leaving the report, the logs and the harness's state archive intact; every completeness check (the manifest's artifact check, the scorer's evidence check, C8's loader) looks only at the steps that are listed, and none checks that every archived state snapshot and the final step appear in the list; in the sealed challenge the historical validity re-derivation also missed its truncated-trajectory case. C17p (stale pages): all five counted it. Every policy-record page served after the change still showed the starting limit although the database row changed; the validity term that checks served pages looks only at steps up to the change, and the certifier predicate that would catch it runs only during certification, not in the classifier (the clinic blocks' re-derivation has the equivalent check). C09a (two committing requests against one approval event): the scorer, the evaluator/classifier and the pipeline scored I2 as passed, because the claims I2 term requires at least one committing request, not exactly one; the baseline, which requires exactly one, caught it. That I2 must fail here is our normative reading, since the pre-registration phrases I2 in terms of approval events. C09b (two approvals in the world): C8, the scorer, the evaluator/classifier and the pipeline rejected it as an invalid intervention, which would replace the episode instead of counting it as an I2 failure: the before-commit term binds a single approval and is undefined for two in all three implementations, so they agree and the block's void gate would not flag it; only the baseline gave the sealed class. Approval is terminal in the task, so a real double approval would need a portal fault. C8 also counted the three panel and delivery corruptions (C02, C02b, C07), which its declared scope, intervention validity only, does not cover; the classifier and the scorer reject all three.

Trust boundary. On the archive as stored, with the archived evaluator verdict not recomputed, the classifier alone counts 16 of the 19 corruptions: it rejects only the three its own archive checks see (C02, C02b, C07), because its validity row reads the archived verdict. For such an archive the backstop is the scorer's and C8's recomputation through the block's void gate, which voids the whole block rather than replacing the attempt; this is inferred from their recomputed verdicts in this run, not run separately on the stored copies.

Crosswalk. An author-written crosswalk maps the 26 validity and corruption classes of the sealed challenge (two pairs of its controls merged) onto five implementations, and cites for every cell the file, line and digest of the code that decides it (CROSSWALK.json); each claims file is the version the claims manifests pin, and the historical re-derivation is the version the sealed challenge ran. The notice-delivery, notice-control and slot classes have no claims counterpart. C8 checks less than the classifier path by design: it re-derives intervention validity only, leaving delivery, outcome and archive completeness to the classifier and the scorer. The clinic prospective re-derivation checks two properties the claims operational path does not (pages served after the change that still show the old value, and notice-shaped page text), and the claims path checks one that neither clinic tool uses (commit-before-change order). Archive completeness is partial in every column.

implementation	checked	partial	not checked	n/a
historical validity re-derivation (sealed challenge's tool)	14	8	1	3
claims validity re-derivation (C8)	10	4	3	9
claims scorer path	15	7	0	4
claims evaluator/classifier path	15	7	0	4
clinic prospective validity re-derivation	14	1	1	10

Limits. The cases may favour classes the authors expected; the baseline was written after reading the pipeline, though its source was sealed before the run and not tuned afterwards. The cases come from six archives of two models and one instance family, and 16 of the 19 corruptions from one Opus OFF-A archive, so the results are per case, not rates. Where the pre-registration is silent, three expectations are normative readings (`V09`, `C09a`, `C17p`), each with its rule text in the sealed file; where the text allows more than one class, every such class is accepted.

The same comparison on both workflows. After the sealed challenge and the check above, we asked what the protocol adds over a fair simple checker on *identical* evidence and scored properties. For clinic we wrote `clinic_baseline.py` (143 lines, standard library), the counterpart of the claims baseline: it asks whether the session row changed exactly once, after its old location was served in the same login session and inside the agent's own recorded request, and whether any served page contradicts the change (on L0, whether nothing changed); whether exactly one commit and one patient row followed the change; whether the booked clinic equals the session's location at the commit (the nearest-slot term is not archived, so it reports fail or unresolved, never pass); and whether the report matches the committed record. It does not check delivery, the arm served, collateral rows, the confirm state or archive completeness. It was developed only on 92 clinic-family run archives that are not sources of any challenge case, sealed (with the table's decision definitions) at 2026-09-24 10:00Z, and run once on the challenge's cases at 10:01Z; nothing was tuned afterwards. Before writing it, its author had read the challenge's expected verdicts, the sealed tools' recorded outputs and the unmodified control archives, but not the mutation code. *Disclosure:* 16 of the 92 development archives were Block BC archives written while that block was running; the development check compared the baseline only with their archived evaluator scores, no number from them appears in this paper, and BC's reading is selected mechanically by its frozen rule. The sealed challenge's joint figure, 14 of 18 corruptions, is the set both validity readings reject; the protocol rows of Table 2 take one reading at a time. The claims baseline's in-scope misses reflect implementation choices a careful checker need not make (nearest-request attribution, timestamp ordering, no event log, counting history rows rather than change requests); the clinic baseline, which reads more raw sources, matched the protocol within its scope. The added value this shows is breadth of checked properties and, on claims, a more robust causal chain than one simple checker; it does not show that a simple checker cannot match the causal chain.

Table 14: **Matched comparison, every decision (post hoc, author-built, not blind).** Validity: controls accepted, validity-property corruptions rejected; outcome: controls scored correctly, outcome-property corruptions with a target invariant flagged; end to end: an episode is rejected if it is not counted, or is counted with a target invariant flagged. n/a: the checker has no such output; FA/FR: false acceptances/rejections. The clinic rows other than the baseline re-tabulate the sealed challenge's recorded first-run outputs (not re-run; the sealed result, Table 8, is unchanged); the protocol rows are derived from them, not a tool that was run. Claims rows: the post-hoc run of Table 13. From `matched_table.json`, pinned by sha256.

checker	validity: controls	validity: corruptions	outcome: controls	outcome: corruptions	end to end: controls	end to end: corruptions	FA / FR
clinic: the sealed challenge's cases (10 controls, 18 corruptions)							
validity re-derivation, implemented	10/10	9/12	n/a	n/a	n/a	n/a	n/a
validity re-derivation, registered	10/10	10/12	n/a	n/a	n/a	n/a	n/a
outcome scorer	n/a	n/a	10/10	6/6	n/a	n/a	n/a
evaluator (archived verdicts)	10/10	0/12	10/10	0/6	10/10	0/18	18 / 0
protocol, implemented (derived)	10/10	9/12	10/10	6/6	10/10	15/18	3 / 0
protocol, registered (derived)	10/10	10/12	10/10	6/6	10/10	16/18	2 / 0
simple baseline	10/10	6/12	9/10	5/6	9/10	11/18	7 / 1
claims: the post-hoc claims-path cases (10 controls, 19 corruptions)							
C8 validity re-derivation	10/10	9/14	n/a	n/a	n/a	n/a	n/a
outcome scorer	10/10	12/14	10/10	4/5	10/10	16/19	3 / 0
evaluator/classifier	10/10	12/14	10/10	4/5	10/10	16/19	3 / 0
pipeline	10/10	12/14	10/10	3/5	10/10	16/19	3 / 0
simple baseline	10/10	3/14	10/10	3/5	10/10	6/19	13 / 0

Fresh cases for a repaired checker and the reconstruction (post-hoc, author-built fresh cases with expectations sealed before their single run). After the per-episode reconstruction of Appendix J had run on the counted archives, we built 24 further cases (9 valid controls, 15 corruptions: new variants of truncation, stale post-change pages, duplicate commits, panel delivery and commit ordering) from 24 counted claims archives, none a base of Table 13, wrote their expected verdicts from the raw start and end state, and sealed the cases, the expectations, the builder, runner and comparer and both new tools at 10:03:30Z on the same day, before any tool ran on any case (a crash check of the repaired checker on an r28 control case preceded the seal). One run at 10:03:34Z used the frozen claims tools as in Table 13, the r28 baseline unchanged, the repaired checker and the reconstruction's verdict. The frozen pipeline again missed truncation, stale pages and access-log duplicate commits and gave the duplicate approval the wrong class; it is not modified for the counted blocks. The repaired checker, which like the baseline orders events by timestamp and reads no prompts or event log, missed the delivery and ordering corruptions the pipeline catches. The reconstruction contains both kinds of check and missed 0 corruptions with 0 false rejections; that shows it implements the checks it names, not an unbiased coverage estimate, and because it reads the same archives as the pipeline, a fault that falsified every copy of the raw record consistently would pass both. In summary (moved from the main text), on these 24 post-hoc, author-built fresh cases with expectations sealed before their single run, the reconstruction missed none of 15 corruptions, the frozen pipeline 7 (truncation, stale pages, access-log duplicate commits) and the repaired raw-log checker 6 (panel delivery, commit ordering). "Fresh" means unused archive bases, not independent or blind evidence: the cases were built by the authors after the fault classes were known.

Table 15: **Fresh sealed cases (post hoc, author-built, not blind; one run).** Expectations and every tool digest sealed (`SEAL.txt`, 10:03:30Z) before any tool ran on any case; run at 10:03:34Z. No case reuses a base archive of Table 13. Cells as in Table 13. The pipeline is the frozen operational path, unchanged; the repaired checker is the r28 baseline plus step-list completeness, post-change page consistency (policy pages and panel) and an explicit single commit; the reconstruction contains both kinds of check. From `FRESH_RESULT.json`, pinned.

case	operational pipeline (frozen)	C8	scorer	evaluator/ classifier	r28 baseline	repaired checker	reconstruction
`V01` clean control (unarmed L0, Sonnet)	met	met	met	met	met	met	met
`V02` inapplicable change control (ON-I X_down, Opus)	met	met	met	met	met	met	met
`V03` inapplicable change control (ON-I X_up, Sonnet)	met	met	met	met	met	met	met
`V04` OFF-A fired, correct approval (Opus, X_down)	met	met	met	met	met	met	met
`V05` OFF-A fired, correct approval (Sonnet, B_down)	met	met	met	met	met	met	met
`V06` valid behavioural failure (stale approval, Sonnet)	met	met	met	met	met	met	met
`V07` ON-A panel control (Opus, B_down)	met	met	met	met	met	met	met
`V08` ON-A panel control (Sonnet, X_up)	met	met	met	met	met	met	met
`V09` valid counted episode whose RESULT is not on its own line (Sonnet)	met	met	met	met	met	met	met
`F15a` truncated trajectory: final done step missing	**missed**	**missed**	**missed**	**missed**	**missed**	caught	caught
`F15b` truncated trajectory: ends before the commit	**missed**	**missed**	**missed**	**missed**	**missed**	caught	caught
`F15c` incomplete trajectory: a middle step missing	**missed**	**missed**	**missed**	**missed**	**missed**	caught	caught
`F17p` state delta lost on one served page	**missed**	**missed**	**missed**	**missed**	**missed**	caught	caught
`F17p` state delta lost on the ON review panel	caught	**missed**	caught	caught	**missed**	caught	caught
`F17p` state delta lost (served pages)	**missed**	**missed**	**missed**	**missed**	**missed**	caught	caught
`F09a` duplicate commit (access log only, 303, 3 s later)	**missed**	scope	**missed**	**missed**	caught	caught	caught
`F09a` duplicate commit (access log only)	**missed**	scope	**missed**	**missed**	caught	caught	caught
`F09b` duplicate commit (world)	**wrong class**	**wrong class**	**wrong class**	**wrong class**	caught	caught	caught
`F02a` ON panel not delivered on one review step	caught	**missed**	caught	caught	**missed**	**missed**	caught
`F02b` ON panel not delivered on any review step	caught	**missed**	caught	caught	**missed**	**missed**	caught
`F02c` ON panel never rendered	caught	**missed**	caught	caught	**missed**	**missed**	caught

Table 15, continued

case	operational pipeline (frozen)	C8	scorer	evaluator/ classifier	r28 baseline	repaired checker	reconstruction
`F08a` commit logged after exposure but before the change (access log only)	caught	caught	caught	caught	**missed**	**missed**	caught
`F08b` commit before change (world order)	caught	caught	caught	caught	**missed**	**missed**	caught
`F08c` commit before change (world order and access log)	caught	caught	caught	caught	**missed**	**missed**	caught
controls met	9/9	9/9	9/9	9/9	9/9	9/9	9/9
corruptions missed	7/15	9/15	7/15	7/15	12/15	6/15	0/15

E RULE-ONLY CASE-DESIGN CHALLENGE (CASES AND EXPECTATIONS SPECIFIED BY GPT-6 WITHOUT THE IMPLEMENTATION)

This is neither the sealed challenge (Table 8) nor the post-hoc author-built checks of Appendix D, and it changes neither result. Design. GPT-6 (ChatGPT, a new temporary chat with memory off) received a brief of verbatim excerpts of the registered rules (the claims pre-registration and the clinic addenda that govern Block BC) and an archive schema (file list, what each log records, one type skeleton per file); the brief held no code, no paper text, no earlier case, class or finding, and no model outcome, and three redactions removed a historical defect label, a defect narrative and changelog identifiers. It returned 56 case specifications (28 per workflow: 20 corruptions and 8 controls each), each with its expected verdict, its rule basis and any accepted alternative; they were parsed without edits, and GPT-6 marked 1 as ambiguous. *Implementation and seal.* The base-selection, exclusion and scoring rules were committed by digest at 2026-09-24 18:44Z, before any answer text had been retrieved (the retrieval log records none until 2026-09-24 18:51Z). The authors then implemented each specification by literal application of its named edits (a specification that could not be applied as written was to be excluded, not reinterpreted) on a copy of a counted prospective archive (claims: Blocks CLM-B and CLM-BS; clinic: Block BC), editing every copy of a fact and recomputing recorded digests. The answer, specifications, cases, expectations and all implementation, run and comparison code were sealed at 2026-09-24 19:00Z; no tool under test had run on any case before then, so no case could be altered to make a tool fail. A second GPT-6 temporary chat—a model, not a human adjudicator—compared every implementation with its specification and judged 36 of the 38 implemented ones faithful; cases it did not confirm were excluded under the frozen admission rule, and the fidelity claim remains the authors'; its decision was recorded at 2026-09-24 19:08Z. The authors checked every admitted expectation against the rule text and disputed none; expectations are copied verbatim and none was edited, and under the frozen rules a dispute would have been reported beside the case without changing any score. The run began at 2026-09-24 19:08Z, used no model and no world, and the tool and case files did not change during it. *Exclusions (pre-declared rules).* Of the specifications, 18 had no base archive meeting their stated requirement: 17 clinic ones (13 need a successful OFF-A episode, and Block BC has none) and 1 claims one; the brief, being blind, did not say which outcomes each level contains, and the fidelity reviewer judged 2 (LC14, LC15) unfaithful; none of these ran. Admitted: 27 claims cases (20 corruptions, 16 of them expected uncounted, and 7 controls) and 9 clinic cases (6 corruptions, all expected uncounted, and 3 controls). *Tools. claims*: combined pipeline (classifier, void gate), evaluator with classifier, claims validity re-derivation (C8), independent scorer, simple raw-log baseline (r28), repaired raw-log checker (r29); *clinic*: combined pipeline (driver row, void gate), evaluator with classifier (refresh), BC validity re-derivation, independent scorer, simple raw-log clinic baseline (r29). The clinic evaluator path re-applies what the harness computes from archived evidence, with the world readouts as archived. Two disclosed deviations: the runner crashed in its own bookkeeping on the case with a deleted trajectory before any claims tool had produced output; the one-line fix changed only that bookkeeping (both runner digests recorded; the tool files are identical across the two invocations), the claims part then ran once, and the clinic part, which had already run once, was not rerun; and the brief described the agent-attributed access log as the same list as the trajectory's, so three cases were implemented on the trajectory's copy with the slice re-derived, which the fidelity reviewer was told and confirmed.

Results and every miss. Table 16 gives each tool; Table 17 every case. The combined pipeline accepted all 10 controls (7 claims, 3 clinic) and, of the 26 corruptions, rejected or flagged 24 (in-

Table 16: **Rule-only case-design challenge, per tool** (cases and expectations specified by GPT-6 from the registered rules and archive schema only; implemented by the authors, fidelity-reviewed by a model, run once). Controls accepted; FR: controls rejected; corruptions rejected, or counted with the expected failing invariant flagged; FA: corruptions counted as clean; scope: counted, with the attacked invariant outside the tool's outputs; errors: no verdict; exact: countedness and class (or invariants) equal GPT-6's expectation. Exact is stricter than clean-versus-corrupt detection: a corruption rejected under a different registered class is non-clean but not exact.

tool	controls	FR	caught	FA	scope	errors	exact
Claims							
combined pipeline (classifier, void gate)	7/7	0	18/20	0	0	**2**	24/27
evaluator with classifier	7/7	0	18/20	0	0	**2**	24/27
claims validity re-derivation (C8)	7/7	0	6/20	**9**	4	**1**	16/27
independent scorer	7/7	0	10/20	**9**	0	**1**	12/27
simple raw-log baseline (r28)	7/7	0	3/20	**14**	2	**1**	12/27
repaired raw-log checker (r29)	7/7	0	3/20	**14**	2	**1**	12/27
Clinic (Block BC)							
combined pipeline (driver row, void gate)	3/3	0	6/6	0	0	0	9/9
evaluator with classifier (refresh)	3/3	0	6/6	0	0	0	9/9
BC validity re-derivation	3/3	0	5/6	**1**	0	0	8/9
independent scorer	3/3	0	5/6	**1**	0	0	8/9
simple raw-log clinic baseline (r29)	3/3	0	1/6	**5**	0	0	0/9

cluding the wrong-class case below), gave no verdict on 2 and counted none as clean: no false acceptance and 2 abstentions, which is not 26/26 detection. It missed three: on KC01, KC08 (missing trajectory, missing required artifact) the classifier alone returned the expected `harness_error`, but the evaluator raised a file-not-found error, so the pipeline reached no verdict; on KC13 (missing required ON review panel) it rejected the case as `invalid_intervention` rather than `observation_not_delivered`, because the evaluator's panel-geometry validity term precedes the delivery row. The claims validity re-derivation counted 9 corruptions as clean (KC06, KC07, KC09, KC11, KC12, KC13, KC14, KC15, KC20); each falls under a countedness row it does not re-derive by design, since it re-derives intervention validity only. The independent scorer counted 9 claims corruptions as clean (KC02, KC03, KC04, KC05, KC06, KC11, KC14, KC15, KC20): it does not read the stop reason, recorded infrastructure errors, read-target violations, per-call model identity, harness text across steps, access-log episode stamps or the arm payload. In the clinic, Block BC's validity re-derivation and the scorer both counted LC07 (supplied-frame failure on silent arm) as clean: the validity script re-derives frame supply from the frame file, which the case left in place, and the scorer's input check gates only rows the driver counted; the operational driver voided the case as expected, and in a block that disagreement triggers the void gate. The simple raw-log baselines counted 14 of the 16 claims corruptions expected uncounted as clean (the repaired checker the same cases) and 5 of the 6 clinic ones; none reads stop reasons, infrastructure errors, read targets, prompts, episode stamps or arm payloads. The clinic baseline accepted every control but never reports the booked-slot term I1 as passing, so it meets no exact expectation.

Limits. Coverage is uneven: the clinic side has 6 corruptions, all expected uncounted, and no outcome-invariant corruption, because every such specification needed a base outcome Block BC does not contain; 19 of the 27 claims cases use the two L0 archives, and every clinic base is an Opus Block BC archive. The results are per case, not rates. The authors chose the brief's excerpts, wrote the implementation and chose the tools, having read the earlier challenges; the implementation was sealed before any tool ran, and a model, not an independent team, reviewed its fidelity. What was blind is exactly the case design: specifications and expected verdicts were written without the implementation, the paper, earlier cases or model outcomes. The controls are a constructed case set, not a false-positive-rate estimate, and claims and clinic counts are never pooled into a rate. GPT-6's expectations are its readings of the registered rules.

Table 17: **Rule-only case-design challenge, every admitted case** (class names are GPT-6's). A: accepted; R: rejected; C: counted with the expected failing invariant flagged; FA: counted as clean; ERR: no verdict; scope: the attacked invariant is outside the tool's outputs; R*: rejected with another class (named under each part). **Bold**: GPT-6's exact expectation not met. Tools: Pipe. the combined pipeline, Eval. the evaluator with classifier, C8 and Valid. the claims and Block BC validity re-derivations, Scorer the independent scorer, Base. the simple raw-log baseline, Rep. the repaired checker. Controls are pooled into one row per workflow ($\times k$: k alike).

case	class	expected	Pipe.	Eval.	C8	Scorer	Base.	Rep.
KC01	missing trajectory	`harness_error`	**ERR**	**ERR**	**ERR**	**ERR**	**ERR**	**ERR**
KC02	provider limit stop	`infra_rejected`	R	R	R	**FA**	**FA**	**FA**
KC03	recorded infrastructure error	`infra_failure`	R	R	R	**FA**	**FA**	**FA**
KC04	harness stop	`harness_error`	R	R	**R***	**FA**	**FA**	**FA**
KC05	unauthorized file read	`harness_error`	R	R	R	**FA**	**FA**	**FA**
KC06	wrong resolved model	`harness_error`	R	R	**FA**	**FA**	**FA**	**FA**
KC07	registered-slot identity mismatch	`harness_error`	R	R	**FA**	**R***	**FA**	**FA**
KC08	missing required artifact	`harness_error`	**ERR**	**ERR**	R	**R***	**FA**	**FA**
KC09	wrong registered start state	`harness_error`	R	R	**FA**	**R***	**FA**	**FA**
KC10	intervention fired at wrong route	`invalid_intervention`	R	R	R	R	R	R
KC11	prompt harness text drift	`harness_error`	R	R	**FA**	**FA**	**FA**	**FA**
KC12	prompt page text differs from served page	`harness_error`	R	R	**FA**	**R***	**FA**	**FA**
KC13	missing required ON review panel	`observation_not_delivered`	**R***	**R***	**FA**	**R***	**FA**	**FA**
KC14	cross-episode access-log stamp	`harness_error`	R	R	**FA**	**FA**	**FA**	**FA**
KC15	access request predates attempt	`harness_error`	R	R	**FA**	**FA**	**FA**	**FA**
KC16	wrong committed amount with matching report	counted, I1 fail	C	C	scope	C	C	C
KC17	approval event wrong predecessor	counted, I2 fail	C	C	scope	C	C	C
KC18	collateral claim mutation	counted, I3 fail	C	C	scope	C	scope	scope
KC19	incorrect final RESULT target	counted, I4 fail	C	C	scope	C	scope	scope
KC20	registered arm mismatch	`harness_error`	R	R	**FA**	**FA**	**FA**	**FA**
KV01–07	7 valid controls	counted, pass	A×7	A×7	A×7	A×7	A×7	A×7

R*: KC04 C8 `infra_failure`; KC07 Scorer `evidence_error`; KC08 Scorer `unresolved`; KC09 Scorer `evidence_error`; KC12 Scorer `evidence_error`; KC13 Pipe. `invalid_intervention`; KC13 Eval. `invalid_intervention`; KC13 Scorer `invalid_intervention`.

case	class	expected	Pipe.	Eval.	Valid.	Scorer	Base.
LC01	provider rejection before action	`infra_rejected`	R	R	R	R	**R***
LC02	application unreadable infrastructure failure	`infra_failure`	R	R	R	R	**FA**
LC04	served review arm differs from requested LIVE arm	`arm_mismatch`	R	R	R	R	**FA**
LC05	intervention fired outside anchor	`invalid_intervention`	R	R	R	R	**FA**
LC06	state delta lost despite fired event	`state_delta_lost`	R	R	R	R	**FA**
LC07	supplied-frame failure on silent arm	`infra_failure`	R	R	**FA**	**FA**	**FA**
LV05–07	3 valid controls	counted, success	A×3	A×3	A×3	A×3	**A**×3

R*: LC01 Base. `invalid_intervention`.

F EXTERNALLY BUILT CHALLENGE (CASES, EXPECTATIONS AND GENERATOR BY AN EXTERNAL MODEL AGENT)

This challenge differs from the rule-only one (Appendix E) in one respect that matters: the authors neither implemented nor reviewed the cases. Constructor and sandbox. One headless Gemini CLI call (`gemini-3.1-pro-preview`, default approval mode; it could neither run a shell nor write files) worked in a sandbox holding only: the four rule and schema files of the rule-only challenge's brief (byte copies, same redactions), 15 byte-verified copies of counted base archives (9 claims, three each from CLM-B, CLM-BS and CLM-BG, including failed episodes; 6 Block BC clinic ones), and an instruction file asking for at least 20 corruptions and 8 unusual-but-valid controls per workflow, a stdlib-only generator that edits every copy of a changed fact and recomputes recorded digests, and an expectation file. The sandbox held no code of ours, no earlier case, class or finding, no paper and no outcome table; a leak check of the instructions, prompt and every base file found no case identifier, tool name or earlier-round label outside verbatim rule text. *Instantiation and seal.* The agent returned the generator and 56 expectations between markers; both were saved byte for byte and the generator ran once, unedited. It built 55 cases; EB11 is excluded because its own generation raised, and no repair round was used. The cases, expectations, transcript, runner and scorer were sealed at 2026-09-25 09:59:06Z, before any tool under test touched any case (the only earlier tool run was a loader smoke test on unmodified base copies). The run began at 2026-09-25 09:59:09Z, used no model and no world; tool and case files did not change during it. The runner is the rule-only challenge's with only its loader adapted, and the scoring functions are that challenge's, copied verbatim.

Results. Table 18 gives each tool. Against the simple raw-log baselines (claims: the r28 baseline; clinic: the r29 baseline) the pipeline caught 13 claims corruptions the baseline missed and the baseline none the pipeline missed; in the clinic 9 against 1 (EB12, whose edit added a key the schema lacks; the baseline scores it only because it never reports the booked-slot term I1 as passing). Every miss: the pipeline counted EA08, EA09, EA19, EA20 (claims) and EB01, EB03, EB10, EB12, EB14, EB16, EB17, EB18 (clinic) as clean. EB14 and EB18 are undisputed gaps: EB14 changed a collateral appointment's status only in the final state rows, disagreeing with the archived I3 pass, and EB18 set the trajectory's event-log target version equal to its source version while the event log and step copies were unchanged; the pipeline does not cross-check these copies. Three claims misses (EA08, EA19, EA20) edit only an archived derived verdict, which the pipeline silently re-derives instead of flagging the disagreement. It gave no verdict on EA01 and EA18 (a deleted trajectory or claims file: the evaluator raised), and rejected EA11, EB05, EB08, EB09, EB19, EB20 under a class other than the registered one (clinic: the void gate's `G0_void` where the rules name the infrastructure or harness verdict). It did not count EA26, EA27, EB27, controls that change one copy of a fact: the scorer reported the disagreement and the void gate voided the case, whereas the agent expected them counted. *Disputes (scores unchanged).* Written after scoring from a mechanical diff of each case against its base: six edit only a derived field that the refresh re-derives or the rules say is never read (EA08, EA19, EA20, EB03, EB10, EB17); three equal their base (EA09, EA28, EB28); three expect an outcome the unchanged evidence contradicts (EA24, EB24, EB25); two add a key the schema lacks (EB12, EB26); two set a stop reason the rules do not register (EB01, EB16); two are controls that change one copy of a fact, against the instructions (EA26, EA27); one is a control the rules leave open (EB27). On the undisputed cases alone the pipeline caught 14/16 claims corruptions with 0 counted as clean and 2 without verdict (baseline 1 and 11), and 11/13 clinic ones with 2 counted as clean (baseline 2 and 10), accepting every undisputed control.

Limits. The design is shallow: one call, one base per workflow (a CLM-B Opus success and a Block BC success) although several were requested, one-to-three-word case descriptions, and no case marked ambiguous; 19 of 56 expectations are disputable. Most corruptions flip a single field the harness records (stop reason, infrastructure errors, read-target violations, episode marker, a policy-input fault or digest) that the operational validity table reads by design; the raw-log baselines were built to audit intervention validity and the I1/I2 terms and do not read those fields, so part of the gap reflects their narrower scope, and a baseline that reads the validity-table fields was not tested. The constructor is an LLM agent, not an independent research team, and Gemini is also a policy under test elsewhere in this paper (CLM-BG, the same model); the authors chose the rule excerpts,

Table 18: **Externally designed and instantiated challenge, per tool** (cases, expectations and generator by a Gemini CLI agent from the rules, schema and archive copies only; generator run unread, cases sealed, each tool run once; the agent's expectations, disputed ones included). Columns as in Table 16: controls accepted; FR: controls not counted; corruptions caught (rejected, or counted with the expected failing invariant flagged); FA: corruptions counted as clean; scope: the attacked invariant is outside the tool's outputs; errors: no verdict; exact: the agent's full expectation met.

tool	controls	FR	caught	FA	scope	errors	exact
Claims							
combined pipeline (classifier, void gate)	6/8	2	14/20	**4**	0	2	14/28
evaluator with classifier	8/8	0	14/20	**4**	0	2	15/28
claims validity re-derivation (C8)	8/8	0	3/20	**12**	4	1	15/28
independent scorer	6/8	2	8/20	**11**	0	1	5/28
simple raw-log baseline (r28)	8/8	0	1/20	**15**	2	2	6/28
repaired raw-log checker (r29)	7/8	1	2/20	**15**	1	2	4/28
Clinic (Block BC)							
combined pipeline (driver row, void gate)	7/8	1	11/19	**8**	0	0	7/27
evaluator with classifier (refresh)	8/8	0	4/19	**15**	0	0	7/27
BC validity re-derivation	8/8	0	6/19	**9**	4	0	17/27
independent scorer	7/8	1	5/19	**14**	0	0	6/27
simple raw-log clinic baseline (r29)	8/8	0	3/19	**15**	1	0	0/27

the base archives, the instructions and the tools. The results are per case on this suite, not detection rates, and claims and clinic counts are not pooled.

G THE CUE-VERSUS-VALUE BLOCKS (BC, BCS, BGC; BG2)

Design. Block BC (ADDENDUM 18; Claude Opus 5, Claude CLI, frames by file read): 16 instances (seeds 42–49, 51–55, 57–59) under FRX-OFF-A, FRX-CUE-A and FRX-LIVE-A in a frozen per-instance order, plus 3 FRX-LIVE-I controls. G1 withholds an interpretation from a confirmatory contrast with fewer than 13 complete pairs; the rescue-opportunity gate is 8 OFF-A failures. Block BCS (ADDENDUM 19; Claude Sonnet 5, Claude CLI, frames by file read) and Block BGC (ADDENDUM 19; GPT-6, Codex CLI, frames inline) run the same instances; BGC adds FRX-OFF-I and the target-relevance contrast to its block-local Holm family. Block BG2 (ADDENDUM 19; GPT-6, Codex CLI, frames inline) runs Block B's cells only if Block BC is dropped. CUE–OFF asks whether adding the cue-and-link panel changes success, and LIVE–CUE whether displaying the current value in it does. The estimand is the effect at the fixed 48-action budget, so it includes the navigation effort the value saves; the design's increase from 10 to 16 triples before any BC rollout is disclosed in Appendix M. States: BC completed; BCS incomplete at the registered experiment cutoff; BG2 fallback not triggered; BGC's final disposition is stated once, below. Admission (ADDENDUM 19): pass. Frame delivery is admitted per channel (§3.3): a model that reads frames through a file-read tool must read the frame it was handed, and one that receives frames inline must transcribe a random code printed on the frame, with the invocation record naming that frame. Reading-rule source mirrored: audit `block_report.py 493abaaeeb4b`. The LIVE–OFF comparisons are supporting descriptive analyses; they support neither a confirmatory replication claim nor a claim that the effect differs from Block B (moved from Section 3.4).

Registered wording, and the narrower interpretation printed. ADDENDUM 18 registered, for R-D, "The Block-B rescue did not reproduce with the value panel on these fresh instances. The rescue is less stable across instances or across the revised interface than Block B suggested." A non-rejected supporting comparison does not establish either statement, so this paper prints the narrower interpretation of Appendix K (template `BC.R-D`); the registered selector is unchanged. The same holds for the registered R-B ("can elicit successful verification"), R-C ("evidence of recovery from both interventions"), R-E and R-G sentences, which are printed in the review-corrected wording.

Block BC (Claude Opus 5). Over the 16 instances (seeds 42–49, 51–55, 57–59), Claude Opus 5 passed OFF-A 0/16, CUE-A 2/16 and LIVE-A 15/16. LIVE versus CUE (primary): rescue 13, harm 0, tie 3 of 16 pairs, net $+0.81$ [0.49, 0.92] (marginal 95 % interval), $p = 0.000244$ (block-local Holm $p = 0.000488$). CUE versus OFF (key secondary): 2/0/14, net $+0.13$ [-0.09, 0.36], $p = 0.5$ (block-local Holm $p = 0.5$). LIVE versus OFF (supporting, descriptive, outside the paper-wide family): 15/0/1, net $+0.94$ [0.64, 0.99]. At the registered block-local level, displaying the current location increased success relative to this cue-and-link panel. The CUE–OFF contrast did not reject; this does not establish absence of a cue effect. Contradiction, salience, and reduced verification effort remain possible explanations. The LIVE–CUE rejection survives the paper-wide Holm analysis over 12 claim-bearing tests (adjusted $p = 0.00293$). The LIVE–OFF comparison is a supporting descriptive analysis; it supports neither a confirmatory replication claim nor a claim that the effect differs from Block B. The LIVE-I controls passed 3/3; nothing is attributed to the panel from unmatched controls. The independent scorer reconstructs appointment availability from archived rows; calendar compatibility is not re-derived but rests on stated design conditions (the generator and state preservation), whose checks are reported separately. Under that condition, every outcome term is resolved and agreeing with the evaluator on 51 of 51 counted rollouts (Table 9). Independent code re-derived this block's validity verdicts (`audit/failure_mining/P3v2/VALIDITY_INDEPENDENT_blockBC.json`), agreeing with the archive on 53 of 53 attempts. Strict recovery occurred in 1 of 16 CUE-A and 15 of 16 LIVE-A runs, and 1 of 16 CUE-A runs followed the panel's link (descriptive; never a reading condition).

seed	FRX-OFF-A	FRX-CUE-A	FRX-LIVE-A	FRX-LIVE-I
42	stale_applied_flagged	stale_applied_flagged	no_commit	pass
43	stale_applied_unflagged	stale_after_cue	pass	pass
44	stale_applied_unflagged	stale_after_cue	pass	pass
45	stale_applied_unflagged	stale_after_cue	pass	
46	stale_applied_unflagged	stale_after_cue	pass	
47	stale_applied_unflagged	pass	pass	
48	stale_applied_unflagged	stale_after_cue	pass	
49	stale_applied_unflagged	stale_after_cue	pass	
51	stale_applied_unflagged	stale_after_cue	pass	
52	stale_applied_unflagged	stale_after_cue	pass	
53	stale_applied_flagged	stale_after_cue	pass	
54	stale_applied_unflagged	stale_after_cue	pass	
55	stale_applied_unflagged	stale_after_cue	pass	
57	stale_applied_flagged	stale_after_cue	pass	
58	stale_applied_unflagged	pass	pass	
59	stale_applied_unflagged	stale_after_cue	pass	

Block BCS (Claude Sonnet 5). Block BCS started but did not complete (incomplete at the registered experiment cutoff). It contributes no inferential result; its rows are listed in Appendix G as a record only. *Execution record (no interpretation).* Block BCS counted 21 of its 51 registered cells in 24 attempts (3 not counted) before its registered launch cutoff, 12:00 UTC on 25 September. No attempt was logged between 19:30 UTC on 24 September and 04:34 UTC on 25 September: an edit to the paper generator had changed the source digest of the sealed reading rule, so the block's launch gate refused to resume it. The rule was restored byte-identically and the block resumed from its frozen search log, re-running no counted row (`REPAIR_r33/BCS_HALT_DEVIATION.md`). The registered terminal rule had made the runner's failure terminal, so the resumption is a disclosed deviation (Appendix M).

seed	FRX-OFF-A	FRX-CUE-A	FRX-LIVE-A	FRX-LIVE-I
42	stale_applied_unflagged	stale_applied_flagged	pass	
43	stale_applied_unflagged	stale_after_cue	pass	
44	stale_applied_unflagged	stale_after_cue	pass	
45	stale_applied_unflagged	stale_after_cue	pass	
46	stale_applied_unflagged	stale_after_cue	revised_on_reexposure	
47	stale_applied_unflagged	stale_after_cue	pass	
48	stale_applied_unflagged	stale_after_cue	no_commit	

Block BGC (GPT-6): final disposition. Block BGC was not run (launch preconditions unmet by the experiment cutoff). It contributes no inferential result; its paper-wide slots are non-rejection placeholders. The terminal disposition recorded before the experiment cutoff lists the unmet launch preconditions: its release manifest, its admission link and its inline-delivery dry run. It therefore has no row in Table 1.

Block BG2 (GPT-6, fallback). Block BG2 was not run (fallback not triggered).

No per-instance record is printed: the block's state is fallback not triggered.

H THE CLAIMS BLOCKS (CLM-B, CLM-BS)

Design. Block CLM-B (PRECOMMIT_CLAIMS; Claude Opus 5, Claude CLI, frames by file read) runs CLM-OFF-A and CLM-ON-A in a counterbalanced frozen order, with CLM-ON-I and CLM-L0 controls, on 10 frozen generated instances (seeds 0–9); Block CLM-BS (PRECOMMIT_CLAIMS ADDENDUM 1; Claude Sonnet 5, Claude CLI, frames by file read) runs the same cells in the same order. The paired instances are stratified by whether the limit rises or falls and whether the change crosses the eligible amount; on a threshold-crossing increase, copying the displayed limit instead of deriving min(eligible, limit) is wrong, which the derived-decision gate uses. Registered wording: PRECOMMIT_CLAIMS words R1 as "The certified evaluation protocol transfers to a second domain with a derived decision, and the review-step intervention increases full-task success". We prospectively instantiate the same protocol on a second workflow with a different authoritative object and a derived decision rule; this tests task-level reuse, not environment-level transfer. This paper's templates are narrower (no transfer claim is printed, the task cannot repair an already committed approval, and a failure includes no commit and budget exhaustion). States: CLM-B completed; CLM-BS completed. The claims task runs on the clinic task's scaffold, harness

and invoker; the claim it supports is task-level reuse of the protocol, not portability, and engineering line counts are not used as evidence for it.

Decision thresholds. As the ledger's mirror of the registered decision function applies them (probed from it, not typed; registered in PRECOMMIT_CLAIMS): a block with more than 6 replacements or failed readiness gives reading R4 and no result; the rescue question is open only if OFF-A fails on at least 5 pairs (else R2); R1 needs $p < 0.05$, more rescues than harms and a minimum number of rescues, which together require at least 6 (else R3); controls are credible only if every L0 control passes and at least 2 ON-I controls pass; the derived-decision gate needs ON-A to pass on at least 2 threshold-crossing-increase pairs. Sample size and seeds are fixed by the block's specification (10 frozen generated instances (seeds 0–9)).

Evidence and checks. Both blocks are read from the frozen bundle in place: its digest list (`063b7ae44e11`) names 29 files, and the ledger fails unless every one verifies. For each block it asserts that the paper's mirror of `claims_readings.decide`, the registered function itself (loaded read-only from its pinned file) on the archived `decide_inputs`, and the reading stored in the result file are identical (CLM-B: registered decide() = mirror = stored reading; CLM-BS: registered decide() = mirror = stored reading); that the generated report prints the same reading; that the result's disposition and inferential eligibility agree with the state the ledger derives; and that its paper-family p (CLM-B 1.00, CLM-BS 0.00781) and Holm slot (#6, #7) are those of Table 26. The copy-limit baseline in each result file is asserted equal to the certification's.

Audits. Independent scorer (4 terms per rollout; gate passed and passed): every term resolved and agreeing with the evaluator on 24 of 24 (CLM-B) and 24 of 24 (CLM-BS) counted rows; the derived-amount term resolved on 24 and 24; unresolved rows 0 and 0; contradictions 0 and 0. Standard-library validity re-derivation: agreeing on 24 of 24 and 25 of 25 attempts (an interrupted attempt has no archive and is re-derived as uncounted); counted rows valid 24 of 24 and 24 of 24. The scorer imports nothing from the evaluator but was written by the same team from the same specification.

Execution. Attempts 24 and 25; uncounted 0 and 1 (interrupted: 0 and 1, at pair9 CLM-ON-A, attempt 0, recorded as operationally interrupted before the archive was written); reserve replacements 0 and 0; step-cap terminations 0 and 0.

Side by side (descriptive; never pooled, no between-model test).

	CLM-B (Opus 5)	CLM-BS (Sonnet 5)	copy-limit shortcut	clinic B / BS (exploratory)
OFF-A success	10/10	1/10	fails 10	0/10 / 0/10
ON-A success	10/10	9/10	—	10/10 / 6/10
for / against / tie	0/0/10	8/0/2	7/0/3	
net [95% CI], method 10	$0.00\ [-0.28, 0.28]$	$+0.80\ [0.37, 0.92]$		$+1.00\ [0.61, 1.00]$ / $+0.60\ [0.20, 0.83]$
p (exact sign test)	1.00	0.00781	0.0156	0.002 / 0.031
reading	R2	R1	R1	
controls; derived gate	credible; 3/3	credible; 3/3	not credible; 0/3	
ON-I; L0	3/3; 1/1	3/3; 1/1		
policy re-read after the change, OFF-A / ON-A	10 / 10	1 / 8		
Back from review, OFF-A / ON-A	8 / 8	0 / 10		
OFF-A stale approvals; stale-approval rescues	0; 0	9; 8		

Moved from the main text (r31). Controls were credible; threshold-crossing increases were met by Sonnet on 3/3, by Opus on 3/3 and by the scripted copy-limit shortcut on 0/3; CLM-BS's 1 interrupted attempt was re-run without spending the reserve. The contrast estimates the total effect of assigning the review-step panel: it does not separate a verification cue, presentation of the current value, the salience of a disagreement with the agent's draft, or reduced lookup effort (the separation Block BC is built for, in the clinic). A diagnostic invariant-completion fraction and an exposure attribution of these rollouts are in Appendix N. Process signals are counts, never reward: a policy re-read is a request for the Policy records page after the change fired; the re-grounding R2 speaks of is claimed only as this count. Of CLM-BS's ON-A failures, 1 approved the correct amount and failed the final-report term.

Erratum (dated 2026-09-24): the stored claims intervals. The claims driver computes its Newcombe interval with the uncorrected correlation numerator (method 8), the defect FREEZE_r21 fixed in `envos.stats`, while labelling it method 10. Its result files therefore carry 0.00 $[-0.28, 0.28]$ (CLM-B) and $+0.80$ $[0.39, 0.91]$ (CLM-BS). This paper prints the registered method 10, computed by the ledger and by the corrected module (`54f6d791e7c4`), which the ledger asserts agree: 0.00 $[-0.28, 0.28]$ and $+0.80$ $[0.37, 0.92]$ (the two methods coincide unless $ad > bc$, with a and d the both-pass and both-fail pairs). No reading depends on the interval.

Block CLM-B (Claude Opus 5). Over 10 sealed instance pairs (seeds 0–9), Claude Opus 5 met the full approval endpoint on 10/10 without the panel and 10/10 with it; of the pairs, 0 favoured the panel, 0 the arm without it, and 10 neither; net 0.00 $[-0.28, 0.28]$ (marginal 95 % interval), $p = 1.00$. Controls: ON-I 3/3, L0 1/1. Without the panel, the agent succeeded on 10/10 instances under silent supersession; the registered rescue-opportunity threshold was not met. The scripted copy-limit shortcut, run on the same cells before any model, is assigned R1 under the same rules. Every outcome term the independent scorer re-derives is resolved and agreeing with the evaluator on 24 of 24 counted rollouts. The standard-library validity re-derivation agrees with the archive on 24 of 24 attempts.

seed	CLM-OFF-A	CLM-ON-A	CLM-ON-I	CLM-L0
0	pass	pass	pass	
1	pass	pass	pass	
2	pass	pass		
3	pass	pass		
4	pass	pass	pass	
5	pass	pass		pass
6	pass	pass		
7	pass	pass		
8	pass	pass		
9	pass	pass		

Block CLM-BS (Claude Sonnet 5). Over 10 sealed instance pairs (seeds 0–9), Claude Sonnet 5 met the full approval endpoint on 1/10 without the panel and 9/10 with it; of the pairs, 8 favoured the panel, 0 the arm without it, and 2 neither; net $+0.80$ $[0.37, 0.92]$ (marginal 95 % interval), $p = 0.00781$. Controls: ON-I 3/3, L0 1/1. The review-step panel increased full-task success at the registered local level. The OFF-A failure categories and observed pre-commit revisions are reported separately (Appendix H). The OFF-A–ON-A contrast rejects under its registered block-local analysis but not under the paper-wide Holm analysis (adjusted $p = 0.0781$ over 12 tests). The scripted copy-limit shortcut, run on the same cells before any model, is assigned R1 under the same rules. Every outcome term the independent scorer re-derives is resolved and agreeing with the evaluator on 24 of 24 counted rollouts. The standard-library validity re-derivation agrees with the archive on 25 of 25 attempts.

seed	CLM-OFF-A	CLM-ON-A	CLM-ON-I	CLM-L0
0	stale_applied	pass	pass	
1	stale_applied	pass	pass	
2	stale_applied	pass		
3	stale_applied	pass		
4	stale_applied	pass	pass	
5	stale_applied	pass		pass
6	stale_applied	pass		
7	pass	pass		
8	stale_applied	pass		
9	stale_applied	fail: correct amount, other term		

I BLOCK CLM-BG: A PRE-REGISTERED NON-ANTHROPIC CONFIGURATION FOLLOW-UP (GEMINI)

What it is. CLM-BG runs the claims design of CLM-B and CLM-BS with one non-Anthropic policy configuration, Gemini 3.1 Pro Preview (`gemini-3.1-pro-preview`). It is registered in ADDENDUM G (sha256 `bf3999dbd602e432` ...), a separate file that cites the frozen PRE-COMMIT_CLAIMS and changes nothing in it. It reuses the registered claims instances, so it is neither a new sample of instances nor a test of vendor identity alone, and it enters no statement about CLM-B, CLM-BS or the clinic blocks.

Registration timing. ADDENDUM G was written after the CLM-B and CLM-BS results (generated 2026-09-24 04:06Z) had been read and after reviews asked for non-Anthropic evidence: the block's existence, its intervention family and the reuse of the claims instances are outcome-informed, and only its rollouts, analysis and interpretation are prospective. It was frozen at 2026-09-24 21:36Z, before the registered hard build cutoff (23:00Z); an RFC 3161 token from an external timestamp authority (FreeTSA) over the digests of the addendum and its freeze record carries 21:36:27Z (registered deadline 23:30Z) and verifies. The first Gemini task request followed at 21:47Z and the last at 23:34Z. Before the freeze no Gemini call carried a claims-task prompt or frame; the engineering calls used synthetic images only and are archived.

Design identity. The ledger asserts that the block specification's slots (10 OFF-A/ON-A pairs, three ON-I cells and one L0 cell), instances, counterbalanced order and 40-step budget equal CLM-B's. The task text, world, trigger, action schema, scorer, validity rules and registered decision function (`claims_readings.decide`, the file pinned for CLM-B and CLM-BS) are the frozen claims ones.

Configuration delta. Only the policy side changes: model and provider; invoker (one stateless REST `generateContent` request per step from the driver process, instead of a fresh Claude CLI session per step); tools (none, instead of the file-read tool only); frame delivery (the frame inline in every request, instead of an instruction to read the frame file, which the Claude policies executed on 357/357 (CLM-B) and 217/297 (CLM-BS) policy steps, counts pinned before any CLM-BG rollout); and pinned generation settings (temperature 1.0, top-p 0.95, top-k 64, thinking level "high", high media resolution), where the Claude CLI's cannot be set. Within Gemini, OFF-A and ON-A share one delivery channel. Every response's model version and a `models.get` capture before and after the block (`3.1-pro-preview-01-2026` both times) record the model's identity; a server-side change under the same strings would not be detectable.

Admission and the key-check incident (disclosed). Six registered gates (identity, statelessness, request content, model and settings, a stub end-to-end run, and a live scripted certification whose policy requests went to a loopback stub, never to Gemini) passed on the final invoker digest. The first live certification stopped at episode 41 of 48: the invoker's own pre-send check refused to send because the API-key pattern matched inside the base64 bytes of a real portal frame (3 of 915 archived certification frames match). No model call or output was involved. The check now blanks the image bytes before the pattern test and still refuses the exact key; because the invoker's digest changed, every admission gate was re-run from scratch on the final digest, and the superseded records are archived and unused.

Execution and audits. One launch made 25 attempts, 24 counted. One ON-A attempt received a provider HTTP 500 at step 22 and was replaced from the reserve under the registered rule (1 of 6 replacements). All 626 task requests are journaled with the digests of their frame and text, which equal the archived files. The independent scorer re-derives all four terms on the 24 counted rows and agrees with the evaluator on 24, with no contradiction and none unresolved; the standard-library validity re-derivation agrees on 25 of 25 attempts and finds all 24 counted rows valid; a separate offline re-derivation from the raw archives passes all 42 of its checks. Registered secondaries: the change fired in OFF-A 10/10, ON-A 10/10, ON-I 3/3 rollouts; the triggered-only sensitivity has 10 pairs, 5 rescues, 0 harms; the OFF-A labels of the rescued pairs are stale approval 5; and 0 of the 603 policy steps of the counted rollouts were provider-blocked or returned an empty reply.

Disposition, reading and multiplicity. The block is complete and inferentially eligible. The registered `decide()`, the ledger's mirror and the stored reading agree: R3, controls credible (L0 1/1, ON-I 3/3); ON-A succeeded on 1/3 threshold-crossing-increase pairs (the derived-decision qualifier applies to R1 only). Net paired difference $+0.50$ [0.12, 0.76] (Newcombe method 10); the copy-limit baseline, reported beside the reading as registered: R1 (7/0/3, $p = 0.0156$; controls not credible). CLM-BG's exact p (0.0625) is a nominal within-block p: the block was added after the twelve-test paper-wide family was frozen, is not in it and carries no family-wise claim. The result file's `paper_family_p` field and the generated report's "Holm slot XV-1" are identifiers only, and no family computation reads them. The Results paragraph (Section 5.1) is ADDENDUM G's item for R3 with credible controls, and the abstract's sentence the matching variant, selected first-

match by the ledger and filled from the result file; the limitations sentence is the addendum's fixed one. No other sentence interprets CLM-BG, and nothing is pooled or tested between models.

Table 19: **The three claims configurations side by side** (descriptive; no pooling, no between-model test, no interval on a between-model difference). Gemini 3.1 Pro Preview received each frame inline in a stateless REST request with no tools; the Claude models were instructed to read each frame file with their file-read tool. Provider, model, invoker and image-delivery channel differ together; rows describe model–invoker configurations, not vendor effects.

	CLM-B (Opus 5)	CLM-BS (Sonnet 5)	CLM-BG (Gemini 3.1 Pro Preview)
OFF-A success	10/10	1/10	0/10
ON-A success	10/10	9/10	5/10
rescues/harms/ties	0/0/10	8/0/2	5/0/5
exact paired p	1.00	0.00781	0.0625
reading	R2	R1	R3
ON-I; L0	3/3; 1/1	3/3; 1/1	3/3; 1/1
frame delivery	Read tool, 357/357 steps	Read tool, 217/297 steps	inline, every request

pair	case	OFF-A	ON-A	ON-A stop
1	B_up	stale approval	no commit	step cap
2	X_up	stale approval	no commit	step cap
3	B_down	stale approval	**pass**	model done
4	X_up	stale approval	no commit	step cap
5	X_down	stale approval	**pass**	model done
6	X_down	stale approval	**pass** (attempt a1)	model done
7	X_up	stale approval	**pass**	model done
8	B_down	stale approval	**pass**	model done
9	B_down	stale approval	no commit	step cap
10	B_up	no commit	no commit	step cap

arm	n	pass	Wilson 95%	evaluator labels	step cap	re-read	back
OFF-A	10	0	[0.00, 0.28]	no commit 1, stale approval 9	1	0	0
ON-A	10	5	[0.24, 0.76]	correct 5, no commit 5	5	6	9
ON-I	3	3	[0.44, 1.00]	correct 3	0	0	0
L0	1	1	[0.21, 1.00]	correct 1	0	0	0

Table 20: **CLM-BG per pair (top) and per arm (bottom).** Evaluator labels of the counted rollouts; "stop" is the ON-A rollout's stop reason; Wilson intervals with $z = 1.96$; "re-read": policy record requested after the change; "back": returned from the review step. Registered diagnostics, descriptive only.

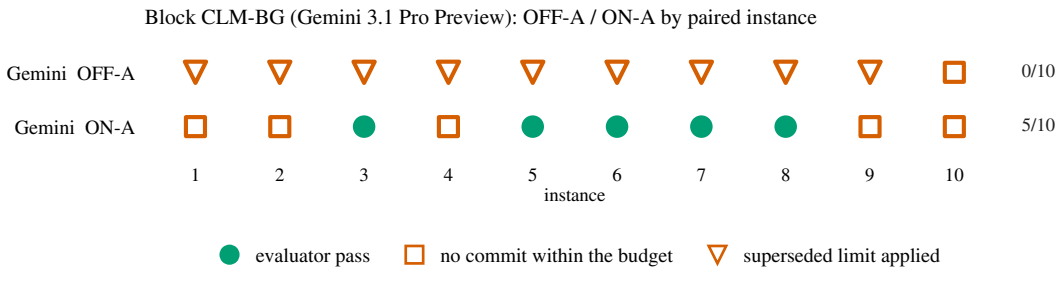

Figure 4: **Block CLM-BG in Figure 3's encoding.** Counted rollouts per paired instance, OFF-A and ON-A; row totals at the right. Gemini 3.1 Pro Preview received each frame inline in a stateless REST request with no tools; the Claude models were instructed to read each frame file with their file-read tool. Provider, model, invoker and image-delivery channel differ together; rows describe model–invoker configurations, not vendor effects.

J THE CLAIMS TASK: INSTRUCTION, INSTANCES, INTERFACE CENSUS AND PER-EPISODE RECONSTRUCTION

Post hoc and author-built, read-only over the frozen claims archives; no episode is re-labelled, re-scored or re-counted, and nothing here changes a registered reading.

Instruction. The step-0 request as sent to the policy (CLM-BS, OFF-A, seed 0; sha256 prefix 18cd71847bed), up to the current page, transliterated to ASCII. It is byte-checked against all 48 counted trajectories and their step-0 requests: header, task text and action schema are identical, and only the claim number varies with the instance. Each step repeats them, then shows the current page's address, title, text (at most 3000 characters) and numbered elements, the recent actions with their memos, and a line naming the screenshot to read.

```
You are a claims adjuster at Meridian Health. You work in a web browser.
The claims portal is at http://portal:8080 -- you are signed in there as marcus.webb.

TASK: Approve claim CLM-78682 in the claims portal. The approved amount is the smaller of (a)
     the eligible amount -- the amount claimed minus the deductible applied, both shown on the
      claim -- and (b) the reimbursement limit on the claim's policy record (Policy records
     page) as it stands when you approve. Enter that amount in the claim's approval form and
     approve the claim. An approval cannot be changed once submitted. When finished, reply
     with exactly one line: RESULT: claim_no=<claim>; approved_amount=<amount>; status=<status
     >


You act in a web browser. Each turn you are shown the current page: its address, its
visible text, and a numbered list of the things you can interact with on it. Reply with
ONE JSON object and nothing else:

{"type":"navigate","url":"http://..."}          load an address
{"type":"click","id":3}                         click element 3
{"type":"fill","id":4,"text":"..."}             type text into element 4
{"type":"select","id":5,"option":"15:00"}       choose an option in a dropdown
{"type":"press","key":"Enter"}                  press a key
{"type":"back"}                                 return to the previous page
{"type":"done","report":"..."}                  finish, with your written report

Any action may also carry "memo":"..." -- a short working note to yourself (at most 300
     characters). Your notes are shown back to you in "What you have done so far".

The element numbers belong to the page shown this turn only; read the new list after
each action before choosing the next one.

Finish with a done action whose report ends with exactly this line:
RESULT: claim_no=<claim>; approved_amount=<amount>; status=<status>

Reply with ONLY the JSON object. No prose, no code fences, no explanation.
```

Instances. Regenerated from the pinned generator, reproducing the sealed seed-table digest, and checked against the start state, end state and event log of every counted episode that used them (Table 21). E is the amount claimed minus the deductible; the correct approval is $\min(E,$ limit at approval$)$, the limit at approval being the new limit on OFF-A and ON-A; the stale approval is $\min(E,$ old limit$)$. The scripted copy-limit policy enters the displayed limit (the old one, or the panel's new one) and never computes the minimum; over the ten pairs it has 10 OFF-A failures, 7 rescues, 0 harms, 3 ties. Controls: ON-I at seeds 0 (1400.00, copy-limit pass), 1 (1550.00, copy-limit pass) and 4 (3856.18, copy-limit fail); L0 at seed 5 (2240.59, copy-limit fail).

Order. The frozen within-pair order held in every pair (Table 22); the split is descriptive, with no test.

Interface census. Every CLI call of the counted episodes (Table 23). The policy ran as a separate OS user with a sealed argument list that allowed Read and denied a named list of tools. Other tools were still requested (names in the table); whether each executed or was refused, and what it returned, was never recorded and is permanently unavailable, so the zero read-target violations constrain Reads only. Errored Reads returned no bytes. Archived and digest-verified on every step: the prompt, the request as sent, the frame, the model's final text and the database state. Never archived: the CLI's system prompt and tool definitions as sent, non-Read tool results, Read text

Table 21: **The claims instances** (seeds of the held-out test split; first: the arm run first in the pair). From `claims_instances.json`, pinned by sha256.

seed	case	first	claimed	deduct.	E	old limit	new limit	correct	stale	copy-limit OFF-A	copy-limit ON-A
0	B_up	ON-A	4006.22	75.00	3931.22	1400.00	3575.00	3575.00	1400.00	1400.00 fail	3575.00 pass
1	X_up	OFF-A	2769.51	75.00	2694.51	1550.00	3050.00	2694.51	1550.00	1550.00 fail	3050.00 fail
2	B_down	OFF-A	1732.65	125.00	1607.65	1325.00	1175.00	1175.00	1325.00	1325.00 fail	1175.00 pass
3	X_up	ON-A	1763.86	175.00	1588.86	725.00	2150.00	1588.86	725.00	725.00 fail	2150.00 fail
4	X_down	OFF-A	4206.18	350.00	3856.18	6725.00	1700.00	1700.00	3856.18	6725.00 fail	1700.00 pass
5	X_down	ON-A	2590.59	350.00	2240.59	3575.00	1400.00	1400.00	2240.59	3575.00 fail	1400.00 pass
6	X_up	OFF-A	3040.59	75.00	2965.59	1325.00	3425.00	2965.59	1325.00	1325.00 fail	3425.00 fail
7	B_down	ON-A	1605.11	175.00	1430.11	950.00	500.00	500.00	950.00	950.00 fail	500.00 pass
8	B_down	ON-A	2404.58	350.00	2054.58	1925.00	725.00	725.00	1925.00	1925.00 fail	725.00 pass
9	B_up	OFF-A	2891.70	350.00	2541.70	1850.00	2225.00	2225.00	1850.00	1850.00 fail	2225.00 pass

Table 22: **Claims outcomes by within-pair order** (descriptive). From `claims_order_outcomes.json`, pinned.

block	order	pairs	OFF-A pass	ON-A pass	for ON-A	for OFF-A	ties
CLM-B (Opus 5)	OFF-A first	5	5	5	0	0	5
	ON-A first	5	5	5	0	0	5
CLM-BS (Sonnet 5)	OFF-A first	5	0	4	4	0	1
	ON-A first	5	1	5	4	0	1

beyond 300 characters, assistant text between tool calls and any thinking, the raw stream and error output, token usage and sampling parameters.

Table 23: **Interface census of the counted claims episodes.** A denied call is a Read whose result is an error; non-Read calls have no archived result. From `claims_interface_census.json`, pinned.

	CLM-B (Opus 5)	CLM-BS (Sonnet 5)
episodes; steps (= CLI calls)	24; 357	24; 297
Read calls; errored (denied) Reads	385; 2	543; 15
Reads naming a file outside the episode's frames (violations)	0	0
steps whose frame Read is confirmed	357/357	217/297
non-Read calls (episodes with any)	21 (15/24)	338 (24/24)
non-Read calls by name	ScheduleWakeup 17, ListAgents 3, ReportFindings 1	ListAgents 182, ExitWorktree 57, ScheduleWakeup 51, ReportFindings 25, navigate 8, CronList 4, click 4, back 3, done 2, fill 1, functions.fill 1
prompt, request, frame, final text, state archived and digest-verified	357/357 steps	297/297 steps

Per-episode reconstruction. `episode_reconstruction.py` (standard library; it imports nothing from the claims pipeline) reads only the raw evidence of each counted episode—world snapshots, the access log, the event log, the arm payload, the delivered prompts and requests with their digests, the step list and the frames—and never an archived verdict field; the search log's success and label are read only for the final comparison. Table 24 counts each property over the 48 episodes and Table 25 gives every episode. In the 46 armed episodes the last old-limit serve preceded the change by 11.6 to 54.5 s (median 19.9) and the change preceded the commit by 22.4 to 199.7 s (median 75.6). The stale-page property has something to check in 31 of them; in the other 15 (9 OFF-A, 4 ON-I and 2 ON-A) no Policy records page was served after the change, and the world row, history row and event log agree that it happened. "Delivered" means that the page text reached the policy and that a frame showing the old value was offered and read on at least one old-limit step (in 1 episode one such frame was not read and another was); the frame Read was not confirmed on 80 of 654 steps overall. Delivery is not belief. *Report-format sensitivity.* One CLM-BS ON-A episode (seed 9) approved the correct amount and failed only the report term, because its single reply line starts with prose before `RESULT:`, which the frozen parser requires at the start of a line. Under a looser reading that accepts `RESULT:` anywhere in the line, the pair would be a further rescue (9 discordant pairs, exact $p = 0.00391$); the registered, stricter reading is the one reported.

Table 24: **Properties reconstructed from raw evidence, over the counted claims episodes** (post hoc, author-built, not blind). n/a: an unarmed L0 episode. From `summary.json`, recounted by the ledger from `episodes.json`.

property	holds	fails	n/a
old limit served before the change (before the commit on L0)	48	0	0
old limit delivered: in the prompt text, and a frame of it read	48	0	0
change fired exactly once	46	0	2
fired inside the agent's own request, in the session served the old value	46	0	2
fired on the first approval-route entry after exposure	46	0	2
rows, history row, event log and arm payload agree (L0: nothing changed)	48	0	0
event log agrees with the access log and the world	48	0	0
panel on every review step of ON arms, on none of OFF-A/L0	48	0	0
no notice text anywhere	48	0	0
one committing request, one approve event, one bound witness	48	0	0
harness snapshots bracket the committing step	48	0	0
old-value serve, change, commit in order (log times and world ids)	48	0	0
pages served after the change show the new limit	46	0	2
pages served before the change show the old limit	48	0	0
step list complete; no agent request outside a step	48	0	0
final done step present, carrying the report	48	0	0
delivered prompt text equals the served page; digests match	48	0	0
reconstructed success and label equal the frozen search log	48	0	0
outcome terms I1, I2, I3, I4 hold in 39, 48, 48, 46 episodes and fail in 9, 0, 0, 2			

Table 25: **Per-episode reconstruction of the counted claims episodes (post hoc, author-built, not blind).** Old limit: steps whose delivered prompt shows the watched row at its old value. Change: seconds from the last old-limit serve to the change (none on L0). Commit: committing requests, and seconds after the change. Pages: Policy records pages served after the change that show the new limit, of all such pages. Panel: review steps carrying the panel, of all review steps. E / limit / approved: eligible amount, governing limit at the commit, approved amount. P/F: invariant passes/fails. Every row also passed every completeness, event-log, panel-flag and delivery check of Table 24, and its reconstructed label equals the frozen search log's unless marked. From `episodes.json`, pinned.

#	block	pair	arm	old limit	change	commit	pages	panel	E / limit / approved	I1–I4	label
1	CLM	pair0	ON-A	3,5	+15.3	1 (+80.1)	2/2	2/2	3931.22 / 3575.00 / 3575.00	PPPP	CORRECT
2	CLM	pair0	OFF-A	3,5	+15.7	1 (+94.6)	2/2	0/3	3931.22 / 3575.00 / 3575.00	PPPP	CORRECT
3	CLM	pair1	OFF-A	3,5	+16.2	1 (+95.8)	2/2	0/2	2694.51 / 3050.00 / 2694.51	PPPP	CORRECT
4	CLM	pair1	ON-A	3	+17.9	1 (+75.7)	2/2	1/1	2694.51 / 3050.00 / 2694.51	PPPP	CORRECT
5	CLM	pair2	OFF-A	3,5	+18.6	1 (+89.2)	1/1	0/2	1607.65 / 1175.00 / 1175.00	PPPP	CORRECT
6	CLM	pair2	ON-A	3,5	+26.9	1 (+67.4)	1/1	2/2	1607.65 / 1175.00 / 1175.00	PPPP	CORRECT
7	CLM	pair3	ON-A	3	+18.4	1 (+81.3)	2/2	1/1	1588.86 / 2150.00 / 1588.86	PPPP	CORRECT
8	CLM	pair3	OFF-A	3	+19.2	1 (+75.8)	2/2	0/2	1588.86 / 2150.00 / 1588.86	PPPP	CORRECT
9	CLM	pair4	OFF-A	3,5	+15.0	1 (+87.3)	2/2	0/2	3856.18 / 1700.00 / 1700.00	PPPP	CORRECT
10	CLM	pair4	ON-A	3,5	+15.3	1 (+60.0)	1/1	2/2	3856.18 / 1700.00 / 1700.00	PPPP	CORRECT
11	CLM	pair5	ON-A	3	+18.5	1 (+80.1)	1/1	2/2	2240.59 / 1400.00 / 1400.00	PPPP	CORRECT
12	CLM	pair5	OFF-A	3,5	+15.6	1 (+83.1)	2/2	0/2	2240.59 / 1400.00 / 1400.00	PPPP	CORRECT
13	CLM	pair6	OFF-A	3	+20.8	1 (+90.2)	2/2	0/2	2965.59 / 3425.00 / 2965.59	PPPP	CORRECT
14	CLM	pair6	ON-A	3,5	+16.9	1 (+73.7)	1/1	2/2	2965.59 / 3425.00 / 2965.59	PPPP	CORRECT
15	CLM	pair7	ON-A	3,5	+15.4	1 (+75.5)	1/1	2/2	1430.11 / 500.00 / 500.00	PPPP	CORRECT
16	CLM	pair7	OFF-A	3,5	+16.4	1 (+85.7)	1/1	0/2	1430.11 / 500.00 / 500.00	PPPP	CORRECT
17	CLM	pair8	ON-A	3,5	+16.4	1 (+63.5)	1/1	2/2	2054.58 / 725.00 / 725.00	PPPP	CORRECT
18	CLM	pair8	OFF-A	3,5	+16.6	1 (+83.8)	2/2	0/2	2054.58 / 725.00 / 725.00	PPPP	CORRECT
19	CLM	pair9	OFF-A	3,5	+19.8	1 (+131.9)	3/3	0/3	2541.70 / 2225.00 / 2225.00	PPPP	CORRECT
20	CLM	pair9	ON-A	3,5	+17.1	1 (+60.8)	1/1	2/2	2541.70 / 2225.00 / 2225.00	PPPP	CORRECT

Table 25, continued

#	block	pair	arm	old limit	change	commit	pages	panel	E / limit / approved	I1–I4	label
21	CLM	onI0	ON-I	3,5	+16.3	1 (+28.3)	0/0	1/1	3931.22 / 1400.00 / 1400.00	PPPP	CORRECT
22	CLM	onI1	ON-I	3	+16.3	1 (+57.5)	1/1	2/2	2694.51 / 1550.00 / 1550.00	PPPP	CORRECT
23	CLM	onI4	ON-I	3	+17.4	1 (+47.0)	1/1	1/1	3856.18 / 6725.00 / 3856.18	PPPP	CORRECT
24	CLM	l0_5	L0	3,6,10	none	1	n/a	0/2	2240.59 / 3575.00 / 2240.59	PPPP	CORRECT
25	CLMS	pair0	ON-A	3	+33.4	1 (+148.5)	1/1	2/2	3931.22 / 3575.00 / 3575.00	PPPP	CORRECT
26	CLMS	pair0	OFF-A	3,5	+11.6	1 (+22.4)	0/0	0/1	3931.22 / 3575.00 / 1400.00	FPPP	STALE
27	CLMS	pair1	OFF-A	3	+32.8	1 (+43.4)	0/0	0/1	2694.51 / 3050.00 / 1550.00	FPPP	STALE
28	CLMS	pair1	ON-A	3,5	+14.1	1 (+156.6)	1/1	2/2	2694.51 / 3050.00 / 2694.51	PPPP	CORRECT
29	CLMS	pair2	OFF-A	3	+29.8	1 (+52.2)	0/0	0/1	1607.65 / 1175.00 / 1325.00	FPPP	STALE
30	CLMS	pair2	ON-A	3	+35.4	1 (+134.2)	1/1	2/2	1607.65 / 1175.00 / 1175.00	PPPP	CORRECT
31	CLMS	pair3	ON-A	3,5,7	+23.2	1 (+115.3)	1/1	2/2	1588.86 / 2150.00 / 1588.86	PPPP	CORRECT
32	CLMS	pair3	OFF-A	3,5	+38.3	1 (+34.6)	0/0	0/1	1588.86 / 2150.00 / 725.00	FPPP	STALE
33	CLMS	pair4	OFF-A	3	+28.6	1 (+57.2)	0/0	0/1	3856.18 / 1700.00 / 3856.18	FPPP	STALE
34	CLMS	pair4	ON-A	3	+27.3	1 (+142.9)	1/1	2/2	3856.18 / 1700.00 / 1700.00	PPPP	CORRECT
35	CLMS	pair5	ON-A	3	+45.7	1 (+138.1)	1/1	2/2	2240.59 / 1400.00 / 1400.00	PPPP	CORRECT
36	CLMS	pair5	OFF-A	3	+24.2	1 (+54.3)	0/0	0/1	2240.59 / 1400.00 / 2240.59	FPPF	STALE
37	CLMS	pair6	OFF-A	3,5,7	+27.8	1 (+32.6)	0/0	0/1	2965.59 / 3425.00 / 1325.00	FPPP	STALE
38	CLMS	pair6	ON-A	3,5,7	+26.7	1 (+199.7)	1/1	2/2	2965.59 / 3425.00 / 2965.59	PPPP	CORRECT
39	CLMS	pair7	ON-A	3,5	+22.3	1 (+64.5)	0/0	2/2	1430.11 / 500.00 / 500.00	PPPP	CORRECT
40	CLMS	pair7	OFF-A	3	+26.6	1 (+71.5)	1/1	0/1	1430.11 / 500.00 / 500.00	PPPP	CORRECT
41	CLMS	pair8	ON-A	3,5	+34.0	1 (+158.1)	1/1	2/2	2054.58 / 725.00 / 725.00	PPPP	CORRECT
42	CLMS	pair8	OFF-A	3,5	+20.0	1 (+50.4)	0/0	0/1	2054.58 / 725.00 / 1925.00	FPPP	STALE
43	CLMS	pair9	OFF-A	3	+41.7	1 (+55.9)	0/0	0/1	2541.70 / 2225.00 / 1850.00	FPPP	STALE
44	CLMS	pair9	ON-A	3	+38.7	1 (+122.3)	0/0	2/2	2541.70 / 2225.00 / 2225.00	PPPF	CORRECT
45	CLMS	onI0	ON-I	3,5,7	+28.6	1 (+26.2)	0/0	1/1	3931.22 / 1400.00 / 1400.00	PPPP	CORRECT
46	CLMS	onI1	ON-I	3,5	+42.7	1 (+46.5)	0/0	1/1	2694.51 / 1550.00 / 1550.00	PPPP	CORRECT
47	CLMS	onI4	ON-I	3	+54.5	1 (+45.6)	0/0	1/1	3856.18 / 6725.00 / 3856.18	PPPP	CORRECT
48	CLMS	l0_5	L0	3,5	none	1	n/a	0/1	2240.59 / 3575.00 / 2240.59	PPPP	CORRECT

K PRE-SCRIPTED INTERPRETATION TEMPLATES

Every sentence the paper can print about a prospective block, as written before any of their outcomes (`paper/readings/templates.tex`, sha256 prefix `af62f98dba4e`; seal status: sealed). The ledger selects templates by the registered rules, re-implemented in `claims.py` and tested against `block_report._bc_readings` and `claims_readings.decide` on synthetic results (rule digest `b23e52dbffcf`, over every function the interpretation depends on). [Name] is a ledger value of the block the template is instantiated for; ‹Model› and its kind are substituted per block. Templates selected in this build are marked. Selected reading ids: BC R-A, A19; BCS INCOMPLETE; BGC NOTRUN; BG2 NOTRUN; CLM-B R2; CLM-BS R1. A wording change after the seal is printed here as a dated erratum beside the sealed text.

Erratum (dated 2026-09-24), template `REUSE.RUN`. "Byte-identical to their clinic-pinned versions" should read "byte-identical to the closest surviving benchmark version": the base the claims worktree was copied from does not survive, so each shared file is compared with the nearest current or dated benchmark version, and the added-line count is an upper bound that also counts benchmark-side drift (printed as "at most"). `BC.S0` Over the [Instances] instances (seeds [SeedList]), ‹Model› passed OFF-A [FRXOffAPassFrac], CUE-A [FRXCueAPassFrac] and LIVE-A [FRXLiveAPassFrac]. LIVE versus CUE (primary): rescue [LiveCueRescue], harm [LiveCueHarm], tie [LiveCueTie] of

[LiveCuePairs] pairs, net [LiveCueNetCI] (marginal 95 % interval), $p =$ [LiveCueP] (block-local Holm $p =$ [LiveCueHolmLocal]). CUE versus OFF (key secondary): [CueOffRescue]/[CueOffHarm]/[CueOffTie], net [CueOffNetCI], $p =$ [CueOffP] (block-local Holm $p =$ [CueOffHolmLocal]). LIVE versus OFF (supporting, descriptive, outside the paper-wide family): [LiveOffRescue]/[LiveOffHarm]/[LiveOffTie], net [LiveOffNetCI].

`BGC.S0T` OFF-I versus OFF-A (target relevance): OFF-I passed [BGC:FRXOffIPassFrac]; [BGC:OffiOffaRescue]/[BGC:OffiOffaHarm]/[BGC:OffiOffaTie], net [BGC:OffiOffaNetCI], $p =$ [BGC:OffiOffaP] (block-local Holm $p =$ [BGC:OffiOffaHolmLocal]).

`BC.S0M` ‹Model› passed OFF-A [FRXOffAPassFrac], CUE-A [FRXCueAPassFrac] and LIVE-A [FRXLiveAPassFrac] on [Instances] instances (Table 26 gives every contrast).

`BC.R-A` **[selected: BC]** At the registered block-local level, displaying the current location increased success relative to this cue-and-link panel. The CUE–OFF contrast did not reject; this does not establish absence of a cue effect. Contradiction, salience, and reduced verification effort remain possible explanations.

`BC.R-B` At the registered block-local level, the cue-and-link panel increased success relative to no panel. The LIVE–CUE contrast did not reject; this establishes neither equivalence nor absence of an additional value-display effect.

`BC.R-B.E` CUE versus OFF rejected in CUE-A's direction at the registered block-local level ([CueOffRescue]/[CueOffHarm]/[CueOffTie], net [CueOffNetCI], block-local Holm $p =$ [CueOffHolmLocal]). The registered rescue-opportunity threshold was not met. The LIVE–CUE contrast did not reject; this establishes neither equivalence nor absence of an additional value-display effect.

`BC.R-C` Both adjacent contrasts rejected in the treatment direction at the registered block-local level; their paired differences and intervals quantify the two increments.

`BC.R-C.E` Both adjacent contrasts rejected in the treatment direction at the registered block-local level (LIVE versus CUE net [LiveCueNetCI], block-local Holm $p =$ [LiveCueHolmLocal]; CUE versus OFF net [CueOffNetCI], block-local Holm $p =$ [CueOffHolmLocal]). The registered rescue-opportunity threshold was not met.

`BC.R-D` The supporting LIVE–OFF analysis did not establish a positive panel effect on these instances. This does not establish absence of an effect or a difference from Block ‹RefBlock›.

`BC.R-E` OFF-A failed on [OffFail] of [FRXOffAN] counted rows; [FRXOffAN] of [OffPlanned] planned rows were observed. Fewer than [OffGate] counted failures were available, so the registered rescue-opportunity threshold was not met. At least one counted OFF-A row succeeded, so the universal OFF-A failure observed in Block ‹RefBlock› did not recur. Post-change source re-reading occurred in [OffReread] of [FRXOffAN] OFF-A episodes.

`BC.R-E.0` OFF-A failed on [OffFail] of [FRXOffAN] counted rows; [FRXOffAN] of [OffPlanned] planned rows were observed. Fewer than [OffGate] counted failures were available, so the registered rescue-opportunity threshold was not met. This threshold statement alone does not establish that universal failure failed to recur.

`BC.R-F.cue` At this budget the cue-and-link panel alone outperformed the value display. No mechanism is inferred.

`BC.R-F.off` The cue-and-link panel reduced success relative to no panel. No mechanism is inferred.

`BC.R-G` Unresolved at the fixed budget. R-G describes the combined cue/value interpretation only and does not negate an independently eligible contrast rejection; each eligible contrast, its direction, interval, and multiplicity status is reported separately. An absent or incomplete contrast is unknown, not non-rejected, and a contrast without discordant pairs has raw $p = 1$: insufficient evidence of a difference, not equivalence.

`BC.ELIG` The ‹Contrast› comparison, which is eligible, rejected in the ‹Dir› direction at the registered block-local level (net [‹C›NetCI], $p =$ [‹C›P]).

`BC.G1` The ‹Contrast› comparison's complete pairs ([‹C›Pairs]) fall below its registered minimum ([‹C›MinPairs]), so it is unknown and receives no confirmatory interpretation. Other eligible contrasts retain their registered analyses.

`BC.A19` **[selected: BC]** The LIVE–OFF comparison is a supporting descriptive analysis; it supports neither a confirmatory replication claim nor a claim that the effect differs from Block ‹RefBlock›.

`BC.GZP` Block ‹Block› has run, but its validity evidence is incomplete ([GzeroWhy]); under its registered validity gate no result is interpreted until that evidence exists.

`BC.VOID` Block ‹Block› is void under its registered validity gate ([GzeroWhy]). It contributes no inferential result; its rows are listed in Appendix G as a record only.

`BC.PEND` Block ‹Block› (‹Model›) had not reported when this version was built. Its arms, instances, contrasts and the interpretations its result will select are fixed in Appendix K; every number it contributes prints as pending until its result file exists.

`BC.NOTRUN` **[selected: BGC]** Block ‹Block› was not run ([Disposition]). It contributes no inferential result; its paper-wide slots are non-rejection placeholders.

BC.INCOMPLETE **[selected: BCS]** Block ‹Block› started but did not complete ([Disposition]). It contributes no inferential result; its rows are listed in Appendix G as a record only.

BC.QI The LIVE-I controls passed [FRXLiveIPassFrac]; nothing is attributed to the panel from unmatched controls.

BC.QE The independent scorer reconstructs appointment availability from archived rows; calendar compatibility is not re-derived but rests on stated design conditions (the generator and state preservation), whose checks are reported separately. Under that condition, every outcome term is resolved and agreeing with the evaluator on [EndpointFull] of [Counted] counted rollouts (Table 9).

BC.QE0 The independent scorer had not been run on this block's counted rollouts when this version was built, so no outcome term is independently re-derived; calendar compatibility rests on stated design conditions in any case.

BC.QV Independent code re-derived this block's validity verdicts ([VindPath]), agreeing with the archive on [VindAgree] attempts.

BC.QV0 No independent re-derivation of this block's validity verdicts exists (it is expected at [VindPath]); its validity verdicts are the evaluator's alone.

BC.QP Strict recovery occurred in [FRXCueARecovered] of [FRXCueAN] CUE-A and [FRXLiveARecovered] of [FRXLiveAN] LIVE-A runs, and [FRXCueALinkFollowed] of [FRXCueAN] CUE-A runs followed the panel's link (descriptive; never a reading condition).

BGC.R-T+ Under this trigger and interface, GPT-6 succeeded more often when an irrelevant session moved than when the target moved. This supports a target-relevance effect on success under the tested trigger and interface; it is not an estimate of a general effect of dynamic environments.

BGC.R-T- GPT-6 succeeded more often under the target change than under the irrelevant one; no mechanism is inferred.

BGC.R-TB GPT-6 failed often even when the target did not move (FRX-OFF-I failed on [BGC:OffIFail] of [BGC:OffIN] counted rows; [BGC:OffiOffaPairs] matched pairs), so its OFF-A failures are not shown to be specific to supersession; its cue and value contrasts are read with that caveat.

BGC.R-TC The target-relevance contrast is unresolved: insufficient evidence of a difference, not equivalence (FRX-OFF-I failed on [BGC:OffIFail] of [BGC:OffIN] counted rows; [BGC:OffiOffaPairs] matched pairs).

BGC.T-X No reading was registered for the target-relevance contrast before the block ran; it is reported as numbers only.

QH.survive **[selected: BC]** The ‹Contrast› rejection survives the paper-wide Holm analysis over [HolmPaperM] claim-bearing tests (adjusted $p =$ [‹C›HolmPaper]).

QH.local **[selected: CLM-BS]** The ‹Contrast› contrast rejects under its registered block-local analysis but not under the paper-wide Holm analysis (adjusted $p =$ [‹C›HolmPaper] over [HolmPaperM] tests).

QH.pending The paper-wide Holm-adjusted p of the ‹Contrast› contrast awaits the rest of the family (it is at most [‹C›BonfPaper]).

BG2.S0 GPT-6 on Block B's instances: OFF-A [BGtwo:FRXOffAPassFrac], ON-A [BGtwo:FRXOnAPassFrac] ([BGtwo:OnOffRescue]/[BGtwo:OnOffHarm]/[BGtwo:OnOffTie], net [BGtwo:OnOffNetCI], $p =$ [BGtwo:OnOffP]).

BG2.OPP-NOT The registered rescue-opportunity gate was not met (OFF-A failed on [BGtwo:OffFail] of [BGtwo:FRXOffAN]).

BG2.OPP-MET OFF-A failed on [BGtwo:OffFail] of [BGtwo:FRXOffAN], which meets the registered rescue-opportunity gate.

BG2.EFF-ON At the registered local level, the review-step panel increased GPT-6's full-task success on Block B's instances.

BG2.EFF-OFF At the registered local level, the review-step panel reduced GPT-6's full-task success; no mechanism is inferred.

BG2.EFF-NONE The paired contrast did not reject: insufficient evidence of a difference, not zero.

BG2.PEND Block BG2, the predeclared fallback that runs only if Block BC is dropped, had not reported when this version was built.

BG2.G1 The paired contrast has [BGtwo:OnOffPairs] complete pairs, fewer than its registered minimum of eight, so it is unknown and receives no confirmatory interpretation; its paper-wide slot is a non-rejection placeholder.

BG2.GZP Block BG2 has run, but its validity or result-agreement evidence is incomplete; under its registered admissibility rule no result is interpreted until that evidence exists.

BG2.VOID Block BG2 is void under its registered validity gate. It contributes no inferential result.

BG2.NOTRUN **[selected: BG2]** Block BG2 was not run ([BGtwo:Disposition]).

BG2.INCOMPLETE Block BG2 started but did not complete ([BGtwo:Disposition]); it contributes no inferential result.

CLM.S0 **[selected: CLM-B, CLM-BS]** Over [OnOffPairs] sealed instance pairs (seeds [SeedList]), ‹Model› met the full approval endpoint on [ClmOffAPassFrac] without the panel and [ClmOnAPassFrac] with it; of the pairs, [OnOffRescue] favoured the panel, [OnOffHarm] the arm without it, and [OnOffTie] neither; net [OnOffNetCI] (marginal 95 % interval), $p =$ [OnOffP]. Controls: ON-I [ClmOnIPassFrac], L0 [ClmLzeroPassFrac].

CLM.R1 **[selected: CLM-BS]** The review-step panel increased full-task success at the registered local level. The OFF-A failure categories and observed pre-commit revisions are reported separately (Appendix H).

CLM.R1-NC A local paired success difference was observed, but the registered control-credibility criterion was not met; no transfer claim is made.

CLM.R1-DG The derived-decision gate was not met ([XupOn]/[XupN] threshold-crossing increases succeeded). This result does not establish performance beyond the scalar-copy shortcut.

CLM.R2 **[selected: CLM-B]** Without the panel, the agent succeeded on [OffPass]/[ClmOffAN] instances under silent supersession; the registered rescue-opportunity threshold was not met.

CLM.R2-H The rescue-opportunity gate was not met. Nevertheless, ON-A reduced full-task success relative to OFF-A ([OnOffRescue]/[OnOffHarm]/[OnOffTie], net [OnOffNetCI], local $p =$ [OnOffP]).

CLM.R3 Without the panel, the agent failed on [OffFail]/[ClmOffAN] instances under silent supersession; a positive panel effect was not established at the registered local level.

CLM.R3-H At the registered local level, ON-A reduced full-task success relative to OFF-A ([OnOffRescue]/[OnOffHarm]/[OnOffTie], net [OnOffNetCI], local $p =$ [OnOffP]).

CLM.R3-NC The registered control-credibility criterion was not met (L0 [ClmLzeroPassFrac], ON-I [ClmOnIPassFrac]), so no transfer claim is made either way.

CLM.R4 The block failed its registered readiness or execution requirements ([RfourWhy]). It provides no admissible model-effect result; all attempts and exclusions remain in the engineering record.

CLM.QCL **[selected: CLM-B, CLM-BS]** The scripted copy-limit shortcut, run on the same cells before any model, is assigned [CopyLimitReading] under the same rules.

CLM.QE **[selected: CLM-B, CLM-BS]** Every outcome term the independent scorer re-derives is resolved and agreeing with the evaluator on [EndpointFull] of [Counted] counted rollouts.

CLM.QE0 The independent scorer had not been run on this block's counted rollouts when this version was built.

CLM.QV **[selected: CLM-B, CLM-BS]** The standard-library validity re-derivation agrees with the archive on [VindAgree] attempts.

CLM.QV0 No independent validity re-derivation of this block exists (Appendix C.1); its validity verdicts are the evaluator's alone.

CLM.QL The claims task shares the scaffold, harness, invoker and vendor with the clinic task: it is a second decision rule for the same intervention, not an independent environment, and no cue or announcement was varied in it.

CLM.PEND Block ‹Block› (‹Model›) had not reported when this version was built; its interpretations and their qualifiers are fixed in Appendix K.

CLM.NOTRUN Block ‹Block› was not run ([Disposition]); it contributes no inferential result, and its paper-wide slot is a non-rejection placeholder.

CLM.INCOMPLETE Block ‹Block› started but did not meet its registered completeness requirements ([Disposition]); it contributes no inferential result.

SHORT.BC ‹Model› on the same instances (Block ‹Block›): OFF-A [FRXOffAPassFrac], CUE-A [FRXCueAPassFrac], LIVE-A [FRXLiveAPassFrac]; block-local Holm $p =$ [LiveCueHolmLocal] (LIVE–CUE) and $p =$ [CueOffHolmLocal] (CUE–OFF); [ReadingShort].

SHORT.CLM ‹Model› on CLM-B's frozen slots (Block ‹Block›): OFF-A [ClmOffAPassFrac], ON-A [ClmOnAPassFrac], $p =$ [OnOffP]; [ReadingShort]. The two models share instances, so their difference is descriptive.

SHORT.STATE ‹Model› on the same instances (Block ‹Block›): [ReadingShort].

FRAG.R-A at the block-local level the value display increased success relative to the cue-and-link panel

FRAG.R-B at the block-local level the cue-and-link panel increased success relative to no panel

FRAG.R-B.E CUE versus OFF rejected in CUE-A's direction at the block-local level; the rescue-opportunity threshold was not met

FRAG.R-C both adjacent contrasts rejected in the treatment direction at the block-local level

FRAG.R-C.E both adjacent contrasts rejected in the treatment direction at the block-local level; the rescue-opportunity threshold was not met

FRAG.R-D the supporting LIVE–OFF analysis did not establish a positive panel effect

FRAG.R-E OFF-A failed on only [OffFail] of [FRXOffAN] counted rows, below the rescue-opportunity threshold

FRAG.R−E.0 OFF-A failed on [OffFail] of [FRXOffAN] counted rows (of [OffPlanned] planned), below the rescue-opportunity threshold

FRAG.R−F.cue the cue-and-link panel outperformed the value display at the block-local level

FRAG.R−F.off the cue-and-link panel reduced success relative to no panel at the block-local level

FRAG.R−G the registered rules leave the combined interpretation unresolved

FRAG.ELIG the eligible ‹Contrast› comparison rejected in the ‹Dir› direction at the block-local level

FRAG.G1 the ‹Contrast› comparison is incomplete and receives no confirmatory interpretation

FRAG.R−T+ success was higher when an irrelevant session moved than when the target moved (block-local)

FRAG.R−T− success was higher when the target moved than when an irrelevant session moved (block-local)

FRAG.R−TB the target-relevance contrast did not reject, and GPT-6 also failed often when the target did not move

FRAG.R−TC the target-relevance contrast is unresolved (not rejected)

FRAG.T−X no reading was registered for the target-relevance contrast

FRAG.PEND pending when this version was built

FRAG.NOTRUN not run ([Disposition])

FRAG.INCOMPLETE started but not completed ([Disposition]); no inference

FRAG.GZP validity evidence incomplete; no result interpreted

FRAG.VOID void under its validity gate; no conclusion

FRAG.CLM.R1 the panel increased full-task success at the registered local level

FRAG.CLM.R1−NC a local paired difference was observed but the control-credibility criterion was not met; no transfer claim

FRAG.CLM.R1−DG the derived-decision gate was not met

FRAG.CLM.R2 OFF-A succeeded often, below the rescue-opportunity threshold

FRAG.CLM.R3 a positive panel effect was not established

FRAG.CLM.HARM the panel reduced full-task success at the registered local level

FRAG.CLM.R4 readiness or execution requirements not met; no admissible result

FRAG.PW.survive surviving the paper-wide Holm analysis

FRAG.PW.local not significant under the paper-wide Holm analysis

FRAG.PW.pending paper-wide Holm pending

APP.pending Its finalized implementation is frozen before each of two pre-registered prospective applications.

APP.pending.draft Its finalized implementation is to be frozen before the first rollout of each of two prospective applications, whose registrations are not yet frozen.

APP.two Its finalized implementation is evaluated prospectively in two workflows, clinic booking and insurance-claim approval.

APP.bc Its finalized implementation is evaluated prospectively in clinic booking, with a second pre-registered application in insurance-claim approval.

APP.clm Its finalized implementation is evaluated prospectively in insurance-claim approval, with a pre-registered application in clinic booking.

APP.none Neither of its two pre-registered prospective applications yielded an admissible result.

ABS.BC.lead In clinic booking, counterbalanced no-panel, cue-and-link and current-value conditions on [BC:InstancesWords] fresh instances (Claude Opus 5)

ABS.BC.PEND had not reported when this version was built.

ABS.BC.NOTRUN were not run ([BC:Disposition]).

ABS.BC.INCOMPLETE started but did not complete ([BC:Disposition]) and support no inference.

ABS.BC.VOID are void under their registered validity gate and support no model conclusion.

ABS.BC.GZP have run, but their validity evidence is incomplete, so no result is interpreted.

ABS.BC.R−C rejected both adjacent contrasts in the treatment direction at the registered block-local level

ABS.BC.R−C.E rejected both adjacent contrasts in the treatment direction at the registered block-local level

ABS.BC.R−A showed at the registered block-local level that displaying the current location increased success relative to the cue-and-link panel, with no rejection of cue versus no panel

ABS.BC.R−B showed at the registered block-local level that the cue-and-link panel increased success relative to no panel, with no rejection of value versus cue

ABS.BC.R−B.E rejected cue versus no panel in the cue's direction at the registered block-local level

ABS.BC.R−F.cue showed at the registered block-local level that the cue-and-link panel outperformed the value display

ABS.BC.R−F.off showed at the registered block-local level that the cue-and-link panel reduced success relative to no panel

ABS.BC.R−D did not establish a positive effect of the value panel over no panel in a supporting descriptive analysis

ABS.BC.R−E found fewer counted no-panel failures than the registered rescue-opportunity threshold requires (OFF-A failed on [BC:OffFail] of [BC:FRXOffAN] counted rows), and at least one counted no-panel row succeeded

ABS.BC.R−E.0 found fewer counted no-panel failures than the registered rescue-opportunity threshold requires (OFF-A failed on [BC:OffFail] of [BC:FRXOffAN] counted rows, of [BC:OffPlanned] planned)

ABS.BC.R−G left the combined interpretation unresolved under the registered rules

ABS.BC.ELIG although the eligible ‹Contrast› comparison rejected in the ‹Dir› direction at the block-local level

ABS.BC.G1 (the ‹Contrast› comparison was incomplete and receives no confirmatory interpretation)

ABS.PW.all and every block-local rejection survives the paper-wide Holm analysis

ABS.PW.none but no block-local rejection survives the paper-wide Holm analysis

ABS.PW.some and only the ‹Contrast› rejection survives the paper-wide Holm analysis

ABS.PW.pending pending the paper-wide Holm analysis

ABS.CLM.lead In insurance-claim approval, where the approved amount is derived,

ABS.CLM.PEND the registered comparison with and without the review-step panel (Claude Opus 5) had not reported when this version was built.

ABS.CLM.NOTRUN the registered comparison with and without the review-step panel was not run ([ClmB:Disposition]).

ABS.CLM.INCOMPLETE the registered comparison started but did not meet its completeness requirements and supports no inference.

ABS.CLM.R4 the registered block failed its readiness or execution requirements and provides no admissible model-effect result.

ABS.CLM.R1 the review-step panel increased full-task success for Claude Opus 5 at the registered local level ($p =$ [ClmB:OnOffP])

ABS.CLM.R1−NC a local paired success difference was observed, but the registered control-credibility criterion was not met; no transfer claim is made

ABS.CLM.R1−DG ; the derived-decision gate was not met, so performance beyond the scalar-copy shortcut is not established

ABS.CLM.R2 without the panel Claude Opus 5 succeeded on [ClmB:OffPass] of [ClmB:ClmOffAN] instances under silent supersession; the registered rescue-opportunity threshold was not met

ABS.CLM.R3 without the panel Claude Opus 5 failed on [ClmB:OffFail] of [ClmB:ClmOffAN] instances under silent supersession; a positive panel effect was not established at the registered local level

ABS.CLM.HARM ; nevertheless the panel reduced full-task success at the registered local level

ABS.AUDIT.hist Independent audits re-derive every validity verdict of the historical blocks ([VindAgree] of [VindAttempts] agree under the rules as implemented) and the booked clinic of every historical booking, but verify no historical success in full

ABS.AUDIT.pending ; prospective coverage is reported per block, and planned evidence collection is not counted as verification.

ABS.AUDIT.some ; on the prospective blocks, with clinic calendar compatibility resting on stated design conditions, every outcome term is resolved and agreeing with the evaluator on [ProspEndpointFull] of [ProspCounted] counted rollouts.

CONCL.fixed These contrasts estimate within-configuration effects; unrejected comparisons do not establish absence or equivalence.

CHAL.PEND A second archive challenge, built by a different model and sealed before any Claude agent read it (seal record [ChallengeSealSha]), had not been run when this version was built.

CHAL.RUN A second, sealed archive challenge built by a different model met [ChallengeMet] of [ChallengeN] frozen expectations on its first and only run ([ChallengeCorruptRejected] of [ChallengeCorruptInstantiated] instantiated corruptions rejected; [ChallengeValidAccepted] of [ChallengeValidInstantiated] valid cases accepted); its [ChallengeMissed] misses are listed in Appendix C as its comparison output words them.

QCONFIG GPT-6 received frames inline; Claude received frames through file reads. Each configuration passed its specified delivery and isolation checks. Within-configuration paired contrasts under a counterbalanced arm order estimate intervention effects. Cross-configuration differences include model, invoker, delivery-channel, and inference-setting differences and do not isolate vendor or model effects.

REUSE.PEND The count of reused and newly implemented components (a deterministic comparison with the clinic-pinned versions) had not been produced when this version was built.

REUSE.RUN [ClmReuseUnchanged] of [ClmReuseShared] shared components are byte-identical to their clinic-pinned versions, the others gained [ClmReuseAddedLines] lines, and [ClmNewFiles] new files ([ClmNewLines] lines) implement the task, its checks and its audits.

L THE PAPER-WIDE MULTIPLICITY FAMILY

A block's execution disposition and inferential eligibility govern every manuscript output: failed-admission, void and R4 blocks contribute no inferential result, their reserved multiplicity slots taking $p = 1$ only as non-rejection placeholders displayed as 'not admissible'; started-but-incomplete blocks are distinguished from unrun ones; and only contrasts meeting their registered completeness and validity requirements are interpreted (moved from Section 3.4). Every claim-bearing test enters one paper-wide Holm family of twelve tests. The family is fixed in the frozen ADDENDUM 19 and never shrinks; 12 of 12 resolved. Its historical slots are retrospective, and the family does not make the exploratory search that preceded those blocks prospective. Readings are selected by the local registered analyses; the paper-wide adjusted p is a qualifier. The claims contrasts were individually pre-registered before rollout; because the paper-wide family was finalized only after those outcomes existed, its adjustment of the claims tests is retrospective and is reported as a sensitivity analysis, not prospective multiplicity control. Block BC's family was frozen before any BC outcome, so its registered block-local Holm adjustment is retained as prospective multiplicity control. The Newcombe intervals are marginal 95 % intervals, not simultaneous family-adjusted intervals, and the paper-wide adjustment does not repair Block BS's eligibility problems (delivery-confounded, Appendix B).

#	block (model)	contrast	role	registered in	status	p	$\min(1, mp)$	Holm (paper)
1	A2 (Opus 5)	S-C vs S-L2	hist. primary	PRECOMMIT S0–S6, ADD. 3	observed (retrospective)	1.00	1.00	1.00
2	B (Opus 5)	OFF-A vs ON-A	hist. primary	ADD. 5	observed (retrospective)	0.00195	0.0234	0.0215
3	BS (Sonnet 5)	OFF-A vs ON-A	hist. primary	ADD. 6	observed (retrospective)	0.0312	0.375	0.281
4	BC (Opus 5)	CUE vs LIVE	primary	ADD. 18	observed	0.000244	0.00293	0.00293
5	BC (Opus 5)	OFF vs CUE	key sec.	ADD. 18	observed	0.5	1.00	1.00
6	CLM-B (Opus 5)	OFF-A vs ON-A	primary	PRECOMMIT_ CLAIMS	observed	1.00	1.00	1.00
7	CLM-BS (Sonnet 5)	OFF-A vs ON-A	repl. of 6	ADD. 19; PRECOMMIT_ CLAIMS ADD. 1	observed	0.00781	0.0938	0.0781
8	BCS (Sonnet 5)	CUE vs LIVE	repl. primary	ADD. 19	not admissible	*not admissible*	*not admissible*	*not admissible*
9	BCS (Sonnet 5)	OFF vs CUE	repl. key sec.	ADD. 19	not admissible	*not admissible*	*not admissible*	*not admissible*
10	BGC or BG2 [BGC] (GPT-6)	CUE vs LIVE, or OFF-A vs ON-A	primary	ADD. 19	not admissible	*not admissible*	*not admissible*	*not admissible*
11	BGC (GPT-6)	OFF vs CUE	key sec.	ADD. 19	not admissible	*not admissible*	*not admissible*	*not admissible*
12	BGC (GPT-6)	OFF-A vs OFF-I	target rel.	ADD. 19	not admissible	*not admissible*	*not admissible*	*not admissible*

Facts that follow from the family before any new result. (1) Block B's paper-wide Holm-adjusted p is at most $m\,p = 0.0234$, below α whatever the other slots hold. (2) Block BS cannot reject under this fixed paper-wide Holm family because A2's $p = 1$ occupies a later rank: its adjusted p is at least 0.0625. (3) Block A2 never rejects. (4) The first Holm step needs $p \leq 0.00417$; with at most 16 pairs (Block BC's size) the favourable splits that pass it are k:0 for $k = 9, \ldots, 16$ and k:1 for $k = 12, \ldots, 15$. (5) Under ordered outcomes, the adjacent gains divide at most 16 favorable discordances. Neither 8–0 contrast passes the first Holm threshold, but both can survive later steps if sufficiently many other family tests reject: with six smaller family p-values rejecting first, two 8–0 contrasts ($p = 0.00781$ each) have Holm-adjusted $p = 0.0469$. Paper-wide support therefore depends on the complete family. An instance favours at most one adjacent contrast (CUE-A must pass to favour CUE–OFF and fail to favour LIVE–CUE), so the reading that both reject under the block-local Holm analysis (R-C) needs at least 7 favourable discordances on one and 6 on the other, 13 in all, with none adverse: with 16 triples it is reachable only if at least 13 of them are favourable. (6) A contrast that rejects under its registered block-local analysis but not under the

Table 26: **Every claim-bearing test, with its local and paper-wide standing.** for/ag./tie: discordant pairs favouring the treatment (for Block A2, the hypothesised failure), favouring the control, and tied; net: paired difference with a Newcombe marginal 95% interval, not family-adjusted; p: exact two-sided sign test. Holm (block): within the block's registered family. Holm (paper): over all twelve tests (12 of 12 resolved; *until every slot resolves: the range the adjusted p can take whatever the unresolved slots hold); †historical: hypothesis-generating (predecessor protocol) and retrospective in the family. A slot whose block is not run or not admissible is a non-rejection placeholder, shown as such. Not in the family: the LIVE–OFF comparisons (descriptive), every control cell, the A2S and BF screens, the sensitivity filters and every cross-model comparison. The for/ag./tie counts are evaluator verdicts; of rows 1–3's 31 paired-arm successes the independent scorer confirms the clinic on 31 and all five invariants on 0. Row 2's rescue reading departs from the registered claim rule (Appendix U).

#	block, model	contrast	role	for/ag./tie	net [95% CI]	p	Holm (block)	Holm (paper)
1†	A2, Opus 5	S-C vs S-L2	hist. primary	1/0/7	-0.13 [-0.47, 0.22]	1	—	1.00
2†	B, Opus 5	OFF-A vs ON-A	hist. primary	10/0/0	$+1.00$ [0.61, 1.00]	0.00195	—	0.0215
3†	BS, Sonnet 5	OFF-A vs ON-A	hist. primary	6/0/4	$+0.60$ [0.20, 0.83]	0.0312	—	0.281
4	BC, Opus 5	LIVE–CUE	primary	13/0/3	$+0.81$ [0.49, 0.92]	0.000244	0.000488	0.00293
5	BC, Opus 5	CUE–OFF	key sec.	2/0/14	$+0.13$ [-0.09, 0.36]	0.5	0.5	1.00
6	CLM-B, Opus 5	OFF-A–ON-A	primary	0/0/10	0.00 [-0.28, 0.28]	1.00	—	1.00
7	CLM-BS, Sonnet 5	OFF-A–ON-A	repl. of 6	8/0/2	$+0.80$ [0.37, 0.92]	0.00781	—	0.0781
8	BCS, Sonnet 5	LIVE–CUE	repl. primary	*not admissible* (incomplete at the registered cutoff)				*not admissible*
9	BCS, Sonnet 5	CUE–OFF	repl. key sec.	*not admissible* (incomplete at the registered cutoff)				*not admissible*
10	BGC / BG2→BGC, GPT-6	LIVE–CUE	primary	*not admissible* (not run; Appendix G)				*not admissible*
11	BGC, GPT-6	CUE–OFF	key sec.	*not admissible* (not run; Appendix G)				*not admissible*
12	BGC, GPT-6	OFF-I–OFF-A	target rel.	*not admissible* (not run; Appendix G)				*not admissible*

paper-wide Holm analysis is printed with exactly that qualification. (7) Every slot of the family is resolved; the adjusted values are in Table 26.

Erratum (dated 2026-09-24): Newcombe intervals of the historical blocks. The historical paired intervals were labelled Newcombe method 10 but computed with the uncorrected correlation numerator, $ad - bc$ (method 8; FREEZE_r21 round 3). All five of them (Table 33, the no-action-cap row included) are recomputed with the corrected module (`envos/stats.py`, `54f6d791e7c4`, imported read-only) and with the ledger's own method-10 implementation, which agree. The correction changes the correlation only when $ad > bc$, which holds for 0 of them, so 0 of the printed intervals change (largest change of any bound below 10^{-9}); the label "method 10" now describes the computation. The claims driver's stored intervals have the same defect (Appendix H).

M DECISIONS TAKEN IN THE FINAL NIGHT, DISCLOSED

Six procedural decisions were taken during the night before the experiment freeze. Each preceded the relevant freeze and every outcome it could affect; each record is pinned by the ledger (`FREEZE_r21/CODE_FINAL.md`, the round-3 pre-images of the spec and the templates, the superseded seal record, the claims block log).

decision	when, relative to outcomes	what changed	inferential consequence
transport timeout	before any BC-family model rollout; certification rerun from scratch	only the Docker transport timeout, 30 s → 120 s, after a certification episode exceeded it under host overload; code refrozen	none: no task rule, threshold, action budget or model setting changed; the claims blocks ran from their own pinned code
branch deadline	before any BC outcome and before the branch rule was frozen	the fallback deadline, 06:00 → 09:00 UTC, because certification and feasibility were projected to finish minutes before it	none: the branch criterion was unchanged
template wording	before any outcome and before the experimental freeze	Two sealed templates said "randomized" for the fixed seeded order; the word became "counterbalanced", the original seal retained and the templates resealed	none: no order assignment changed
an interrupted attempt	during CLM-BS, before that attempt had a valid archive or an outcome	One process terminated by the operator session; under the frozen crash-recovery rule the world baseline was verified, the attempt logged as interrupted and the cell rerun	none: no outcome existed to discard, and the reserve was not spent
BC design size	before BC's freeze and any BC rollout	10 → 16 triples, after exact-test arithmetic showed one registered two-contrast interpretation unreachable at 10; the original design is retained in the record	a registered interpretation becomes reachable; no BC outcome informed the change, and no extension is permitted after rollout begins
second-vendor admission	after the original GPT-6 admission failed on a criterion specific to the Claude file-read channel; no GPT-6 task block ran before quota exhaustion	admission amended to define an inline-frame channel separately	no GPT-6 model result is reported, and neither admission attempt is used as model evidence (configurations and the per-channel rule: Appendix G; status: pass)

Block BCS: halt and resumption (a disclosed deviation). An operator error in a paper file, not a rollout outcome, halted Block BCS for about nine hours: an edit changed the source digest of the sealed reading rule, and the block's launch gate refused to resume it. Under the registered terminal rule the runner's failure was terminal; instead the rule was restored byte-identically and the block resumed from its frozen search log, re-running no counted row, with its cells, order and cutoff unchanged. The operator had seen the verdicts of its first four attempts; the resumption depended on none of them. The block ended incomplete at its registered experiment cutoff, so both of its paper-wide slots enter as $p = 1$ placeholders (Appendix G).

N Diagnostic invariant-completion fraction and exposure attribution of the claims blocks

Beside binary success we describe the claims blocks by two diagnostics built from existing evidence only—the archived independent-scorer records and validity terms, read through the ledger; no rollout was run for this table, and nothing in it is a test or a reading. *Diagnostic invariant-completion fraction*: of the independent scorer's 4 outcome terms per counted rollout, the fraction met, pooled over an arm's rollouts. This diagnostic fraction is specific to EnvOS's endpoint decomposition and is not comparable to partial scores reported by other computer-use benchmarks. *Exposure attribution* (labels after OSWorld 2.0 (Yuan et al., 2026)): *exposure* means the archived validity record shows the target limit changed after the superseded value had been delivered to the policy and before the commit. *Handled*: exposed, and the approved amount at commit is the one derived from the limit then in force. *Failed after exposure*: exposed, amount wrong. *Untested*: the arm moves no target record (ON-I moves an irrelevant one, L0 nothing), or exposure is not evidenced. A handled rollout can still miss the full endpoint on another term, which the fraction shows. For Sonnet 5 the fraction was 30/40 (0.75) without the panel and 39/40 (0.98) with it, and its OFF-A rollouts were 9 failed-after-exposure and 1 handled.

O The policy interface compared with other harnesses

Our policy interface is not intended to reproduce OSWorld-style native computer use. It is a controlled semantic browser harness: each step exposes the current screenshot together with clipped visible page text and an indexed element list, accepts a fixed browser-action schema, and provides only self-authored working memory through a bounded memo/history channel. This reduces low-level perception and pointer-control variance so the experiment can isolate response to state changes. It is

Table 27: **Claims blocks: diagnostic invariant-completion fraction and exposure attribution** (counted rollouts; ledger-derived from the archived audit records; descriptive only, not comparable to other benchmarks' partial scores).

model	arm	n	full endpoint	invariant-completion fraction	Handled	Failed after exposure	Untested
Opus 5	OFF-A	10	10	40/40 (1.00)	10	0	0
Opus 5	ON-A	10	10	40/40 (1.00)	10	0	0
Opus 5	ON-I	3	3	12/12 (1.00)	0	0	3
Opus 5	L0	1	1	4/4 (1.00)	0	0	1
Sonnet 5	OFF-A	10	1	30/40 (0.75)	1	9	0
Sonnet 5	ON-A	10	9	39/40 (0.98)	10	0	0
Sonnet 5	ON-I	3	3	12/12 (1.00)	0	0	3
Sonnet 5	L0	1	1	4/4 (1.00)	0	0	1

more instrumented than a pure pixel benchmark but also more constrained: a fixed action schema, no shell, a navigation allowlist, an isolated user, bounded memo/history and an action cap (Table 28). The structured text and element list may raise absolute task success relative to pixel-only computer use, while the stateless per-step process may lower it relative to an agent with persistent conversational memory; we therefore avoid ranking the harness globally as easier or harder. Our results are fixed-instance intervention contrasts within one interface under the stated assumptions (§3.4), not directly comparable leaderboard scores: because EnvOS estimates matched intervention contrasts under one fixed interface rather than absolute computer-use capability, cross-benchmark score comparability is not required for them; interface transfer remains an external-validity question.

Table 28: **Interfaces compared.** EnvOS beside OSWorld 2.0 (Yuan et al., 2026) and WebArena-style browser benchmarks (Zhou et al., 2024). Representative interfaces used in the cited evaluations; this table is not an exhaustive description of every agent configuration those benchmarks support. The paired arms of an EnvOS block share every row of this table, so the interface cancels within a contrast; it does not cancel across benchmarks.

	observation	action	memory	budget	runs, scoring, purpose
EnvOS	screenshot + clipped page text + indexed element list	fixed browser-action schema; no shell; allowlist; sealed, isolated user	self-authored memo ($\leq$300 chars), last 48 actions	48 / 40 actions	paired arms per instance, one run per cell; binary endpoint re-derived independently; matched intervention effects
OSWorld 2.0	screenshot only	Claude: native computer-use tool; others: pyautogui code	agent-dependent	500 steps, 3 s pause per action	one run per model; binary and checkpoint partial score; exposure attribution; end-to-end capability
WebArena-style	structured browser state (accessibility tree)	browser actions	agent-dependent	per benchmark	task success; web-task capability

P THE PROTOCOL AS A MINIMAL ABSTRACTION, AND CONTINUOUS CHANGE

The abstraction. Stripped of the testbed, an EnvOS intervention is a tuple (v, E, τ, c, V, I, R): an authoritative variable v on which the correct action depends; an exposure predicate E over the archived request and frame record (the old value of v was served in a frame delivered to the policy); a trigger τ that applies the transition of v inside the first eligible request after E holds, in one transaction with its event record; a matched inapplicable control c that fires the same trigger on a variable irrelevant to the correct action; a validity table V mapping each attempt's archived evidence to counted or not counted before its outcome is scored; invariants I evaluated on authoritative state at commit time and on the agent's report; and an interpretation map R from registered test outcomes to pre-written sentences, frozen with V and I before rollout. The checks of Section 3 test these components: certification tests τ and c with scripted policies; mutation batteries and the archive challenges test V and I; the independent scorer and validity re-derivation re-implement I and V; and the claim ledger executes R. Nothing in the tuple names the semantic-browser interface. What depends on it is how E is evidenced (here, request logs and delivered text frames; in a screenshot-only harness, the frame that rendered the value), and portability to another harness is untested (Section 7).

Asynchronous and continuous change. The tuple assumes a discrete transition. For state that changes asynchronously or continuously (a price feed, a queue position), the same structure applies when the environment versions v: E becomes "version k was served", τ publishes version $k{+}1$ on the first eligible request after that, and I scores the commit against the version current at commit time, so that a commit acting on version k is distinguishable from one acting on a later version. A continuously drifting quantity additionally needs a decision-relevant threshold crossing (as the claims task's threshold-crossing instances use) or a stated tolerance, so that the correct action changes discretely. Without versioned authoritative state neither the exposure certificate nor the commit-time scorer can be built, and the protocol does not apply. We have not implemented this extension.

Q TASK TEXT AND GENERATOR

The instruction of instance seed 0 (split `test_id`) as frozen in §4 of the pre-registration (sha256 prefix `cdce6a1d` of the frozen block; transliterated to ASCII below). Every arm of one instance runs the same bytes: on the archived trajectories of Block A2, 8 of 8 instances carry one instruction sha256 across all their rollouts (7 distinct texts over the block; two seeds may draw the same patient, label and date). The release manifest pins, separately, the task module's default instruction text (`66260588`), not a per-instance text.

```
Two things. First, open Marie Lefevre's appointment APT-20448 in the booking portal and note
     its status. Then book Grace Adeyemi into the influenza vaccination session on Thursday 20
      August 2026 at 15:00, using appointment number APT-20451. Her contact email is in her
     intake message in the clinic mailbox (Mailpit); her calendar for that day is in Nextcloud
      -- look at it for a clash before you choose the time. The portal's Clinic sessions page
     lists where each session runs. If 15:00 is unavailable, book the nearest later available
     slot. Then confirm the appointment in the portal.
Finish your reply with exactly one line in this form:
RESULT: appointment_no=<...>; clinic=<clinic>; slot=<HH:MM>; status=<status shown in the
     portal>; note=<APT-20448 status>
```

Generator. One salted digest of (seed, split) draws every element of an instance: the patient, the distractor appointment, the session date and the control session two days later, the appointment number, the intake mail and calendar entry, and the ordered (old, new) clinic pair from a pool of four names. The pair is assigned by a Latin-square rule, so each of the twelve ordered pairs occurs once in twelve consecutive seeds and the dropdown position of the old and the new clinic is counterbalanced across the block. The neutral work — a distractor record to read, a mailbox lookup and a calendar check — is drawn from (seed, split) alone, so it is identical for every level of an instance by construction. The certificate check `filler_facts_identical_across_arms_and_clinic_neutral` scans the intake message and the calendar entry for any clinic name and the frozen cue tokens, on the module's own instance (`S-train-0`, in all 10 certificate files); regenerating the 18 evaluated instances, 0 name a clinic. The check does not cover the distractor appointment: that record carries a clinic from the same pool, on 6 of the 18 instances the new one (23 counted rows), which is how the report-flag rule came to read the distractor's location as a flag (ADDENDUM 17; Table 34). Five salted splits (train, development, held-out test, compositional, sealed) partition the seed space. Three uses of a test instance are distinct here. *Generation*: every instance is generated from a held-out test seed. *First-model evaluation*: a family's first model runs instances no model had run (Block A2: seeds 8–15; Block B: seeds 20–29; Block BC: seeds 42–49, 51–55, 57–59). *Later-model replication*: Blocks A2S, BS and BF re-use those instances (A2S a subset of A2's, BS all of B's, BF 4 of B's), so their instances are known to the experimenter, not fresh. A seed whose treatment outcome has become known is burnt for any new first-model block.

R POLICY INTERFACE AND INVOKER SPECIFICATION

This appendix states the harness-authored part of the agent–environment interface of the P3v2 experiments as the archived rollouts were run. It is not the whole interface: what was not archived (the CLI's built-in system prompt and tool definitions, its decoding defaults, per-step element lists and page titles, the results of non-`Read` tool calls) is marked *unavailable* where it arises. It is derived from the source files whose sha256 the release manifest of each block pins, not

from the current tree. Table 29 lists the pinned digests. Of the pinned files, 19 of 20 (block, file) pairs are available byte-for-byte. Each of the 82 rollouts in the five search logs records the digests of the code it ran. All match the block manifest except as follows: 4 rollouts of Block BS (1 of them counted) ran harness `2439d3a38953`, the version before that block's manifest. They ran under the block's first manifest (`9aa433099a46`), which the dated amendment applying ADDENDA 11 and 12 rebuilt as `ead3fa67655a`; each row records the digests it ran, but no file with the first manifest's digest survives. That version is also available and is listed below. The harness versions used (`2439d3a38953`, `b9f4fdb5b8e4`, `fe4b9bb76d40`) differ only in the read-target check and in delivery bookkeeping. Their prompt constants and the functions `build_prompt`, `_history_line`, `with_agent_notes` and `_trim_note` are AST-identical. The two browser versions differ only in the port configuration: observation, clipping and rendering are identical. The two task-module versions that are available differ only in the delivery-certification function (Addendum 16). The instruction, generator, step budget and history window are identical. †The task-module version used by Block A2 (`17f6c7cd72f2`) was not in the working tree when this specification was derived from the pinned sources (Block A2's task module (`17f6c7cd72f2`), absent from the working tree, was recovered from a backup and verified by its digest, and ships in the supplement). For A2 the instruction text is taken from the archive instead (item 1). This specification was derived before that module was recovered, so for A2 it describes the module's checkpoint and delivery-certification logic as it then stood only through the archived checkpoints, and does not restate its declared history window (item 4). Repository paths are written relative to `/envos`.

Table 29: Pinned code identity per block (sha256, first 12 hex). "CLI" is the policy CLI binary.

Block	requested model	step cap	harness	browser	invoker	task module	CLI
A2	`claude-opus-5`	48	`2439d3a38953`	`657b2381dfe2`	`99c919f1fca4`	`17f6c7cd72f2`†	`15e2d05148f8`
A2S	`claude-sonnet-5`	48	`fe4b9bb76d40`	`1c1010d38219`	`095a8a4081a8`	`5c38ce9c2153`	`15e2d05148f8`
B	`claude-opus-5`	48	`2439d3a38953`	`657b2381dfe2`	`99c919f1fca4`	`5c38ce9c2153`	`15e2d05148f8`
BS	`claude-sonnet-5`	48	`b9f4fdb5b8e4`	`657b2381dfe2`	`99c919f1fca4`	`5c38ce9c2153`	`15e2d05148f8`
BF	`claude-fable-5-1`	48	`fe4b9bb76d40`	`657b2381dfe2`	`095a8a4081a8`	`5c38ce9c2153`	`15e2d05148f8`

(1) Policy prompt, verbatim. Each policy step is *one fresh, stateless* CLI process that receives the text below as its single user prompt. There is no `-continue` or `-resume`, so no conversation is carried from one step to the next. Placeholders are in angle brackets. The `RESULT` line of the action specification is the instance form built by `result_form_for_task`, and the `memo` paragraph is inserted by `with_agent_notes` because the task module declares `AGENT_NOTES = True`. The last sentence is appended by the invoker, and `<FRAME PATH>` is the step's screenshot inside the per-episode observation directory (`/srv/envos-obs/<16 hex>/step_NN_before.png`). Non-ASCII characters are transliterated: the em dash becomes `-`.

```
You are an office assistant for a clinic. You work in a web browser.
The booking portal is at http://portal-health:8080 -- sign in there as scheduler with the
     password sched2026.

TASK: <INSTRUCTION: see below>


You act in a web browser. Each turn you are shown the current page: its address, its
visible text, and a numbered list of the things you can interact with on it. Reply with
ONE JSON object and nothing else:

{"type":"navigate","url":"http://..."}          load an address
{"type":"click","id":3}                         click element 3
{"type":"fill","id":4,"text":"..."}             type text into element 4
{"type":"select","id":5,"option":"15:00"}       choose an option in a dropdown
{"type":"press","key":"Enter"}                  press a key
{"type":"back"}                                 return to the previous page
{"type":"done","report":"..."}                  finish, with your written report

Any action may also carry "memo":"..." -- a short working note to yourself (at most 300
     characters). Your notes are shown back to you in "What you have done so far".

The element numbers belong to the page shown this turn only; read the new list after
each action before choosing the next one.

Finish with a done action whose report ends with exactly this line:
```

```
RESULT: appointment_no=<number>; clinic=<clinic>; slot=<HH:MM>; status=<status>; note=<status
    of <distractor label>>

Reply with ONLY the JSON object. No prose, no code fences, no explanation.

CURRENT PAGE
address: <page URL>
title: <page title>
--- page text ---
<page text>
--- end page text ---
--- things you can interact with ---
<ELEMENT LIST>
--- end ---

What you have done so far:
<history line for step 0>
<history line for step 1>
...

Your next action:

The current screen is the image at <FRAME PATH>. Read it first.
```

The <INSTRUCTION> is the instance's instruction (Appendix Q). It is archived verbatim in every trajectory. For 82 of the 82 archived model rollouts, including all of Block A2, it equals, byte for byte, the text that the generator `generate_variant_params(seed, split)` produces today. The CLI also prepends its own built-in system prompt and tool definitions. That text is not part of the harness and was not archived, so it is *unavailable*. The CLI version that supplied it is pinned (item 5).

(2) Action schema. This is the JSON object listed in the prompt above. The schema allows 7 action types, with an optional `memo` on any action. A reply is parsed by removing code fences and then taking the *first* brace-delimited substring (non-greedy `\{.*?\}`). That substring must be a JSON object with a `type` key. A reply that does not parse ends the episode (`unparsed_reply`), and so does a `done` action. Element ids index the list shown on that same step, and the browser resolves them positionally. Across the 2548 archived model steps, the actions carried these keys outside the schema: none. The harness ignores extra keys, but they are echoed back in the history. Each action has a 15 s timeout. The episode is capped at 48 policy steps in every block.

(3) Observation per step. Each step delivers two channels, both captured *before* the action:

- **Text**, inside the prompt. It carries the page URL and title, and the page's `document.body.innerText` clipped to 3000 characters (the overflow becomes `... [N more chars]`). Table cells are tab-separated. The text was clipped on 0 of 2548 archived steps. The prompt then lists the interactive elements, at most 400 across frames and shadow roots, one per line as `[id] role "label"`. A line may add `options=[...]` (at most 40), `value="..."` (at most 200 characters) or `href=...`. Labels are clipped to 200 characters, and a table row's own text is appended to its controls' labels (at most 120 characters). The archived pages had a median of 8 elements and a maximum of 172. The per-step element lists and page titles were not archived. Only their count is, so a rendered element block cannot be reproduced from the archive.
- **Image**. This is a PNG screenshot of the whole virtual desktop, taken by the desktop's control server, at 1920x1080 px (82 runs). It reaches the model only if the model calls the CLI's `Read` tool on <FRAME PATH>. The archive records this per step. The frame was read on 2305 of 2548 model steps (per model in Table 30).

The page settles adaptively before each observation. The page signature is polled every 0.1 s until 5 consecutive polls are unchanged. The wait is never shorter than 1.0 s and never longer than 5.0 s. Navigation is restricted to the three task origins (`http://portal-health:8080`, `http://mailpit:8025`, `http://nextcloud`).

(4) Memo and history semantics. The memo is part of the action. After parsing, it is truncated to its first 300 characters. The full raw reply is archived untruncated. Nothing is scored from a memo. The history block lists the last 48 executed actions, one line

each: `step i: <json.dumps(action)> -> ok=<True|False|None> @ <last 46 characters of the URL after the action>`. The history window (48) is not smaller than the step cap (48), so *every* earlier action and *every* earlier memo is echoed back at every step. The agent's memory is therefore the full list of its own memos and actions, not only the latest memo. It carries no earlier page text or image. `json.dumps` escapes non-ASCII characters in the echo (e.g. `\u2014`). The done step is not echoed, because the episode ends there. Example: the first history line of the Opus FRX-ON-A instance-20 episode (285 characters):

```
step 0: {"type": "navigate", "url": "http://portal-health:8080", "memo": "Priya email: priya.
    sharma@example.org (from Mailpit intake). Next: login portal, check APT-20448 status,
    sessions page, then Nextcloud calendar for Thu 20 Aug 2026."} -> ok=True @ http://portal-
    health:8080/login
```

In the archive, 2476 of 2548 model actions carried a memo. 126 were cut at 300 characters, and 0 archived memos differ from the truncated raw reply. The archived Block A2 episodes had at most 39 steps, so the echo there was complete under any window of at least 38 lines. The harness default is 40 and the later task module declares 48, but A2's own module value is *unavailable*.

Table 30: Per requested model, over all 82 archived model rollouts: policy steps; steps whose frame the model Read; actions with a memo; memo length median/max (characters); memos truncated at 300; wall time per policy call median/max (s).

model	steps	frame read	memo	memo len	truncated	call s
`claude-fable-5-1`	133	133	128	271 / 300	20	8.4 / 17.3
`claude-opus-5`	1148	1145	1110	224 / 300	88	6.3 / 24.9
`claude-sonnet-5`	1267	1027	1238	128 / 300	18	14.1 / 226.1

(5) Invoker settings and model identity. All 2548 archived model steps record the sealed invoker (2548 steps). Each step runs the pinned CLI binary under `sudo -n -u <policy-user> env -i HOME=<policy-home> CLAUDE_CONFIG_DIR=<policy-home>/.claude PATH=/usr/local/bin:/usr/bin:/bin TERM=dumb LANG=C.UTF-8`. The policy user is unprivileged and can read only the observation directory. The arguments are:

```
<cli> -p <prompt> --model <model-id>
  --allowedTools Read
  --disallowedTools Bash BashOutput KillShell Write Edit MultiEdit NotebookEdit Glob Grep
     WebFetch WebSearch Task Agent TodoWrite ToolSearch Skill Workflow SlashCommand Monitor
     SendMessage Artifact EnterWorktree CronCreate RemoteTrigger DesignSync
  --permission-mode default
  --setting-sources ''  --strict-mcp-config
  --output-format stream-json --verbose
```

Each call has a 300 s timeout. A failed call or a provider error is retried on the same prompt and frame after 20 s and again after 60 s. A provider usage limit is never retried: it ends the episode as infrastructure. **Decoding, reasoning and output-length settings** are not passed: the argument list sets no temperature, effort, thinking budget, maximum tokens, maximum turns or system-prompt flag, and no environment variable is set beyond those listed. They are therefore the CLI's and the provider's defaults for the model id. Those default *values* were not recorded and are *unavailable*. Token usage was not recorded for these calls (0 steps with usage). Within one call the model may call tools several times: `claude-fable-5-1` made more than one call on 1 step and none on 0; `claude-opus-5` made more than one call on 17 steps and none on 3; `claude-sonnet-5` made more than one call on 912 steps and none on 237. **Tool calls.** The argument list auto-approves only `Read` and denies 25 tools by name. The model also emitted calls to other tool names: `claude-fable-5-1` 1 call on 1 step (`ReportFindings` 1); `claude-opus-5` 6 calls on 6 steps (`ScheduleWakeup` 5, `ReportFindings` 1); `claude-sonnet-5` 1591 calls on 801 steps (`ListAgents` 812, `ScheduleWakeup` 315, `ReportFindings` 184, `ExitWorktree` 168, `CronList` 52, `navigate` 38, `select` 6, `done` 4, `fill` 4, `click` 3, `Bash` 2, `Navigate` 1, `PushNotification` 1, `back` 1). The invoker archives the *name* of every tool call, but a *result* (executed or refused) only for `Read` calls (`claude-fable-5-1`: 133 ok; `claude-opus-5`: 1 error, 1155 ok; `claude-sonnet-5`: 81 error, 3602 ok). Whether each of the non-`Read` calls was refused or executed is therefore *unavailable* from the archive. The isolation that bounds them is the operating-system account: it can read only the observation directory, and it has no repository, run-directory or docker access. **CLI version.** Every block's manifest

pins the binary's sha256 `15e2d05148f8`. The file on disk today has the same digest. Its name is `claude-2.1.274`, and run offline it reports `2.1.274 (Claude Code)`. **Model identity.** Requested ids: `claude-fable-5-1`, `claude-opus-5`, `claude-sonnet-5`. The evidence of the served model is the `model` field of the CLI's `system/init` stream event, recorded on every step. It equals the requested id on every step of every archived rollout: `claude-fable-5-1`: 133 of 133 steps; `claude-opus-5`: 1148 of 1148 steps; `claude-sonnet-5`: 1267 of 1267 steps. The per-episode label `policy_resolved_model_id`, which is derived from those init events, agrees on all 82 rollouts. The init field is the CLI's own report of the model it was configured to call. It is not a server attestation. Server-side identity evidence (the `model` field of each API response) was not parsed or archived, so it is *unavailable*.

(6) The FRX-ON-A panel, as rendered. On FRX-ON-A (and FRX-ON-I), the portal's review-and-submit step appends one table below the draft, rendered from the *live* `clinic_session` row for the selected service and date. The table has a title, a header row and one data row. It carries no sentence: no statement that anything changed and no explicit verification instruction. It does place the current location beside a draft naming the old one, and that visible disagreement may itself cue a re-check (Appendix T). Its template is `templates/wizard.html`, pinned in the manifests (`bb26ee02bca1`), with the title taken from the deployed spec. Every page that carried the panel in the 25 archived ON rollouts (FRX-ON-A: 19, FRX-ON-I: 6) matched 1 template once the session id, date and location were abstracted. A scan for 16 cue words (e.g. *moved, changed, updated, please, check, verify*) found 0 hits in the panel text. The panel's text channel, with cells tab-separated as in the prompt:

```
Session details for the selected date
Session    Service    Date    Location
<session id>    Influenza    <session date>    <location>
```

Below is the full page text of the review step, as the policy received it on step 20 of the Opus FRX-ON-A instance-20 episode (names are the generator's fictitious ones). The panel is the last table before the buttons, and it shows the new location while the draft above still names the old one:

```
Northgate Health -- Vaccination Scheduling
Signed in as Clinic Scheduler
All appointments New submission Clinic sessions Switch user
1. Patient details
2. Clinic and time
3. Review and submit
Review and submit
Date    2026-08-20
Appointment number    APT-20455
Clinic    Northgate Main
Contact email    priya.sharma@example.org
Notes    None
Patient full name    Priya Sharma
Time slot    15:00
Vaccine    Influenza
Documents    0 attached
Session details for the selected date
Session    Service    Date    Location
SES-0820-FLU    Influenza    2026-08-20    Riverside Annexe

Submit Back
```

S OBSERVABILITY, SENSITIVITY AND THE WITHIN-EPISODE TIMELINE

Table 31: **What the agent could know, and when.** Per condition: whether the authoritative row changes, what announces it, where the current value is visible after the change, and the scripted evidence that the condition is solvable and discriminating — a legal policy that scores 1.0 and a declared deficient policy that does not. The trigger column is the certified anchor: the change fires on the agent's own request, after it has been served the sessions page. Derived from the certificates, the audit spec and the blocks' feasibility artifacts; no cell is typed.

level	change	announced	current value visible at (trigger)	legal policy (1.0)	deficient policy
`S-L0`	none	–	sessions page (unchanged) (–)	`reference`	–
`S-C`	control	modal notice	sessions page; notice (the agent's first / after exposure)	`reference`	`over_apply (0.8)`
`S-L2`	applicable	modal notice	sessions page; notice (the agent's first / after exposure)	`reference`	`stale_location (0.8)`
`FRX-OFF-A`	applicable	none	sessions page (re-read) (the agent's first `/new/` after exposure)	`late_reread_ref`	`reference (0.8)`
`FRX-ON-A`	applicable	none	sessions page (re-read); review step (the agent's first `/new/` after exposure)	`late_reread_ref`	`reference (0.8)`
`FRX-OFF-I`	control	none	sessions page (re-read) (the agent's first `/new/` after exposure)	–	–
`FRX-ON-I`	control	none	sessions page (re-read); review step (the agent's first `/new/` after exposure)	`late_reread_ref`	–

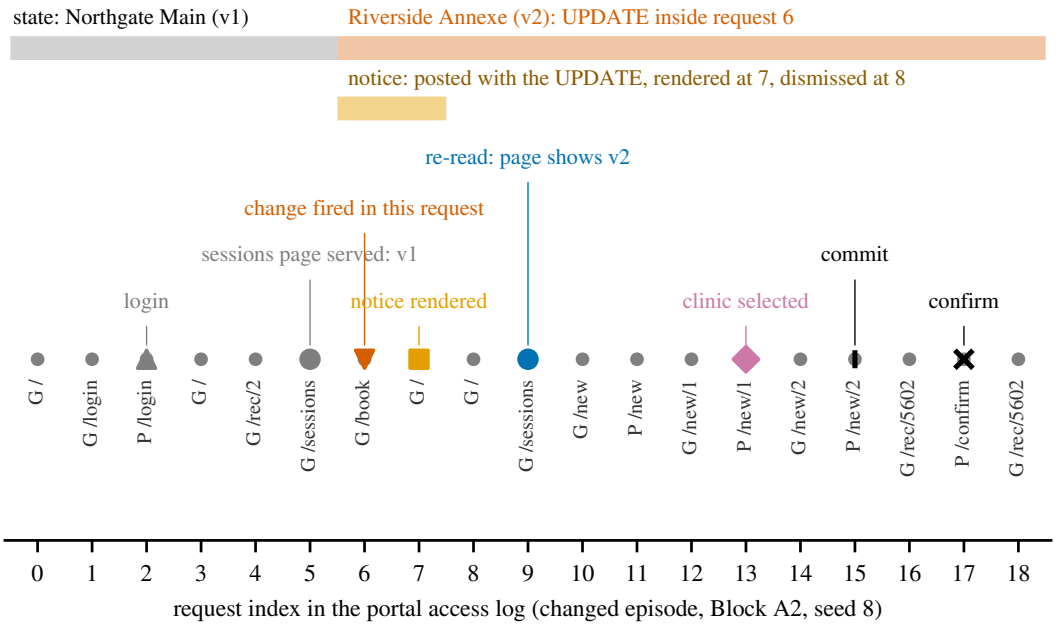

Figure 5: **State precedes channel.** One changed episode (Block A2, seed 8) on the axis of the agent's own requests to the portal. The authoritative row changes inside the request the exposure condition selects, the notice is posted with the same mutation and rendered on the next request, and the agent's re-read of the sessions page shows the new location before the clinic is selected and committed. Every marker is read from the episode's access log, event log and checkpoints.

Table 32: **Sensitivity of the paired results to analysis choices** (historical blocks; exploratory, predecessor protocol). Each block is shown as reported under the rules actually implemented (which counted Sonnet's 3 FRX-ON-A rows whose panel reached it as text only; ADDENDUM 5 would have voided and replaced them), then with rows whose intervention never reached a delivered observation in text or image removed (the ADDENDUM 16 definition), then with action-cap rows removed, then both; a filtered row takes its pair partner with it. For Block BS a last row applies ADDENDUM 5's frame rule as registered, voided rows left unobserved (Appendix B). Deletion is not the registered replacement procedure, whose outcomes are unobserved. Cells count each arm over the rows an analysis keeps; the exact test uses the complete pairs only, so in the ADD. 5 row the control cell also holds the rows whose partner that rule voids ("unpaired"). Computed by `audit/sensitivity_analysis.py`, whose reported column is checked against each block's result file (Block B yes, Block BS yes, Block A2 yes).

block	analysis	complete pairs	control cell	treatment cell	exact p
B	reported (implemented rules)	10	`FRX-OFF-A` 0/10	`FRX-ON-A` 10/10	0.00195
	delivered, text or image (ADD. 16)	10	`FRX-OFF-A` 0/10	`FRX-ON-A` 10/10	0.00195
	no action-cap rows	10	`FRX-OFF-A` 0/10	`FRX-ON-A` 10/10	0.00195
	both filters	10	`FRX-OFF-A` 0/10	`FRX-ON-A` 10/10	0.00195
BS	reported (implemented rules)	10	`FRX-OFF-A` 0/10	`FRX-ON-A` 6/10	0.0312
	delivered, text or image (ADD. 16)	9	`FRX-OFF-A` 0/9	`FRX-ON-A` 6/9	0.0312
	no action-cap rows	8	`FRX-OFF-A` 0/8	`FRX-ON-A` 5/8	0.0625
	both filters	8	`FRX-OFF-A` 0/8	`FRX-ON-A` 5/8	0.0625
	ADD. 5 frame rule, voided rows unobserved (retro.)	7	`FRX-OFF-A` 0/10 (3 unpaired)	`FRX-ON-A` 6/7	0.0312
A2	reported (implemented rules)	8	`S-C` 8/8	`S-L2` 7/8	1
	delivered, text or image (ADD. 16)	8	`S-C` 8/8	`S-L2` 7/8	1
	no action-cap rows	8	`S-C` 8/8	`S-L2` 7/8	1
	both filters	8	`S-C` 8/8	`S-L2` 7/8	1

Table 33: **Paired statistics.** T / C: pairs in which only the treatment / only the control arm succeeded. Net difference P(treatment)−P(control) over matched instances, ties in the denominator, with a Newcombe (1998, method 10) hybrid-score marginal 95% interval (not exact, not family-adjusted); exact two-sided sign test on the discordant pairs. The envos `effect` field of the result files is the discordant-pair *proportion*, which is not a net effect (it reads 0.125 for Block A2, whose net is −0.13). Holm is a *retrospective* correction over the 3 pre-declared primary tests, which were registered at $\alpha = 0.05$ each; it is superseded by the paper-wide family of Table 26 and kept for comparison. The last row drops the pairs in which either arm hit the action cap (retrospective). Block A2S's test was withdrawn before any rollout (ADDENDUM 13); its row is descriptive. All rows are historical and exploratory (predecessor protocol); the intervals are recomputed under the dated erratum of Appendix L.

block	treatment vs control	pairs	T / C / tied	net [95% CI]	exact p	Holm
A2	S-L2 vs S-C	8	0 / 1 / 7	−0.13 [−0.47, 0.22]	1	1
B	FRX-ON-A vs FRX-OFF-A	10	10 / 0 / 0	+1.00 [0.61, 1.00]	0.00195	0.00586
BS	FRX-ON-A vs FRX-OFF-A	10	6 / 0 / 4	+0.60 [0.20, 0.83]	0.0312	0.0625
A2S	S-L2 vs S-C	4	2 / 0 / 2	+0.50 [−0.10, 0.85]	0.5 (descr.)	–
BS, no action-cap pairs	FRX-ON-A vs FRX-OFF-A	8	5 / 0 / 3	+0.63 [0.17, 0.86]	0.0625 (retro.)	–



Figure 6: **Block A2, per instance and within episode.** (a) Outcomes per generated instance: filled, pass; open, fail (the failure's signature code is written under it and spelled out in the key); "1st" marks the arm the seeded order ran first. (b) For each changed run, the action index at which the change fired, the notice frame was delivered, the agent re-read the sessions page, selected the clinic in the wizard, and committed; the shaded segment spans change to commit. 7 of 7 successful runs re-read the authoritative page at or after the request that fired the change and before selecting (the page served by the firing request already carries the new location, so the re-read and the change can coincide); the one failure did not. Dotted line: the 48-action budget.

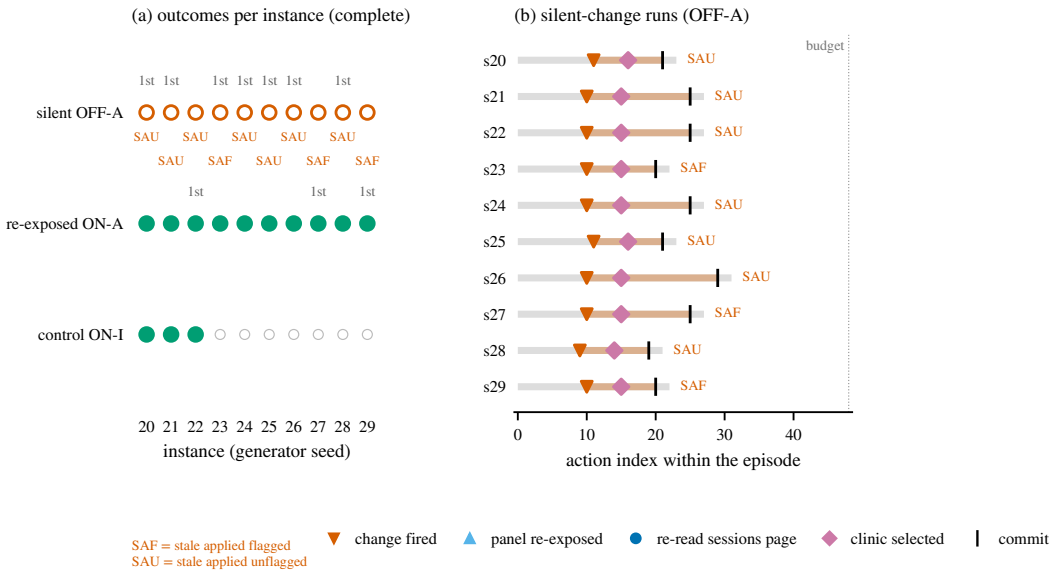

Figure 7: **Block B: silent supersession, with and without re-exposure** (Claude Opus 5, 10 paired instances). Same encoding as Figure 6; the panel marker on FRX-ON arms is the review step re-rendering the live row. FRX-OFF-A 0/10, FRX-ON-A 10/10, $p = 0.002$.

T HISTORICAL, HYPOTHESIS-GENERATING EVIDENCE (PREDECESSOR PROTOCOL)

Every block in this section ran under a predecessor version of the protocol (success-first counting, an unenforced panel-delivery rule; Appendix B); none is pooled with the prospective blocks, and the paper's causal conclusions do not rest on them.

Silent change and the value panel (Block B). Historical Block B motivated the prospective experiments but is not treated as confirmatory evidence under the finalized protocol. Its evaluator recorded a 10/10 paired change in booked-clinic correctness between no-panel and value-panel conditions, and archive re-scoring independently verifies the booked-clinic term on all ten pairs; however, the block did not satisfy one preserved pre-registered claim condition, so we report the contrast as hypothesis-generating historical evidence rather than as a confirmatory rescue claim. Its successes are evaluator passes whose booked clinic is independently verified; none is verified on the full endpoint, because the historical archives do not hold the slot occupancy the nearest-slot term needs (0 of 10; Appendix C.1). The claim condition, the post-outcome sentence that read the contrast from the primary test alone, and the alternative explanations the block cannot separate—among them cueing a re-check versus delivering the value, the question Block BC was built for—are in the supplementary provenance file (Appendix U). Descriptively, no OFF-A run of the silent blocks requested the sessions page after the change (0/24), although a legal scripted policy that does so solves 10 of 10 OFF-A instances.

Other predecessor blocks. A Sonnet 5 block on Block B's instances, a Fable 5.1 screen, an announced-change block whose result did not reproduce an earlier failure of ours (Block A2), the exploratory corpus behind the re-verification hypothesis and their sensitivity analyses are reported, with their limitations, in the supplementary provenance file (Appendix U) and Appendices S and V.

U HISTORICAL RECORDS AND REGISTRATION PROVENANCE (SUPPLEMENTARY FILES)

The historical blocks are motivation and provenance only (Appendix T); none carries a claim of this paper. Their per-cell rates, the per-episode tables of the announced (A2, A2S) and silent (B, BS, BF) predecessor blocks with the detailed accounts moved there from earlier versions of this paper (Block A2's non-replication, Block B's report labels and registered claim rule, and the alternative explanations of its panel effect), and the full registration chronology with its errata are therefore no longer typeset. They ship as a generated supplementary file (`supp_provenance/EnvOS_historical_provenance.tex` and its PDF, listed with their digests in `supp_provenance/MANIFEST.json`), which the claim ledger writes from the same pinned artifacts, macros and generated tables as this paper, so every number in it is a ledger entry. For orientation: the supersession family's pre-registration (`PRECOMMIT_P3v2.md`) is frozen by sha256 (`c280ce46`) with 19 dated addenda below the freeze marker, and the claims pre-registration is a file of the claims blocks' frozen evidence bundle (digest list `063b7ae44e11`); both ship as files, byte for byte. Registered-versus-implemented rules are reconciled in Appendix B, and the final prospective specifications are in Appendices G–H.

V EXPLORATORY CORPUS (STACK B)

The corpus was produced on a separately built stack with its own harness, and is reported in Appendix T as the source of the organising hypothesis, never as a model-level claim. Its structure, re-derived by the claim ledger from the corpus's per-arm and per-rollout files (`audit/failure_mining/STACKB/`, where the corpus's own ledger script agrees on every count): 32 arms across three environments (a clinic, a hotel and a travel desk), of which 27 are hand-written phenomena — among them a selected resource taken between selection and commit, a write silently altered after the commit, a fact revised with no notification, an entity merged under a new name, a policy reissued off the agent's path and an acknowledgement lost — and 5 are controls (clean baselines and cosmetic changes). Claude Opus 5 ran all 32 arms, 93 rollouts in all, from 1 to

7 per arm (24 arms at three); Claude Sonnet 5 ran 23 arms, 22 of them also run on Opus 5, so the two models together cover 33 arms.

Validity ledger. Four properties keep it exploratory, and each is a reason no rate from it is reported. (i) No arm carries a certificate in the sense of Section 3.2: the stack's own test suites do not establish that the change fired once, inside the agent's own request, after the fact it supersedes was served. (ii) Its triggers are hand-coded: heuristics on the agent's own actions (a page load followed by a delay, a view or reservation of the target, a submit) or, in the travel desk's cold-start arm, the episode's start, never certified against the value the agent was served, so an episode in which the agent never saw the superseded value is indistinguishable, from the outcome alone, from one in which it did — the ambiguity the exposure-conditioned trigger of §3.1 removes. (iii) Several cells hit a turn cap, and capped episodes were handled inconsistently (counted as failures in some cells, voided and re-run in another), so a denominator does not uniformly separate "acted on a stale belief" from "ran out of turns". (iv) The corpus's verifier was corrected during its own life, and its versions are not recorded: every archived Opus 5 run record carries a re-score field (93 of 93), of which 2 record a stored failure re-scored as a pass, and a later fix to the hotel task's verifier turned some Sonnet 5 failures on that task into passes and is reported by the corpus's own document, not re-derived here, to leave that task's archived Opus 5 rollouts unchanged. Per-arm pass counts are therefore left to the corpus's own document; the ledger re-derives only the structure above, from files whose digests it pins.

W FULL DEFECT TABLE

The defects include a restarted block in which 3 of 12 episodes never triggered the change, a provider session-limit reply scored as a model decision, and a classifier that would have voided 25 model failures of the historical silent blocks; some showed only in model evidence, which no scripted policy produces.

The defects divide by what caught them: design-time gates, validity rules on a block's first rows, and archived model evidence alone, because no scripted policy produces them (an unauthenticated first request, a dismissal from the sessions page, a report naming the distractor's clinic, and a panel read as text alone). No fix was validated by re-running a model: classification, the checkpoints and the signature are pure functions of the archive, so every corrected row was re-derived from it with the previous value kept beside the new.

11 defects with archived evidence. For 7 of them an archived search-log row scored under the defect already carried a model verdict (`success` or `model_failure`): they would have been reported as model outcomes. The other 4 were caught before any model-scored row existed under them (at Block B's design or certification time, or on a not-counted verdict after its first rollout). The last column counts the archived rows scored under each defect and, in brackets, those with a model verdict; the count 7 is derived from that column, not from the order of this table.

#	defect	archived evidence	rows (model)
1	no working memory across stateless per-step calls; the clean control failed and the stop rule halted the block	`P3v2/_void_blockA_attempt1_2026-09-17/`; `P3v2/_void_blockA_attempt1_2026-09-17/search_log_blockA.json`; `PRECOMMIT_P3v2.md`	1 (1)
2	a provider session-limit reply was scored as a model failure	`P3v2/_void_blockA_attempt1_2026-09-17/`; `P3v2/_void_blockA_attempt1_2026-09-17/search_log_blockA.json`; `PRECOMMIT_P3v2.md`	1 (1)
3	the trigger depended on the agent revisiting a page, so episodes were not instances of the phenomenon	`P3v2/_superseded_blockA_2026-09-17/`; `P3v2/_superseded_blockA_2026-09-17/search_log_blockA.json`; `PRECOMMIT_P3v2.md`	11 (8)

4	the scorer located the change one request early, so the exposure check read false on correct episodes	P3v2/_superseded_blockA_2026-09-17/; P3v2/_superseded_blockA_2026-09-17/search_log_blockA.json; PRECOMMIT_P3v2.md	3 (3)
5	the silent arm as first frozen defeated every policy that re-reads before the wizard, including the certified reference	PRECOMMIT_P3v2.md	–
6	six pairs could reject only if every pair crossed one way	PRECOMMIT_P3v2.md	–
7	the classifier read a silent arm's absent perception checkpoint as 'not certified' and would have discarded every silent-arm failure as invalid	PRECOMMIT_P3v2.md	–
8	the perception checkpoint dropped a dismissal made from a page its route table did not name, so announced-arm failures dismissing there would have been discarded as invalid (found by the independent scorer on the model rows)	P3v2/FINAL_REPORT_blockA2_ERRATUM.md; P3v2/reward_audit_runs/REWARD_AUDIT_defect8_detected_2026-09-21.json; P3v2/search_log_blockA2.json; PRECOMMIT_P3v2.md	5 (5)
9	the read-target validity rule voided episodes on Reads that obtained nothing (errored reads, /dev/null), so a model with that habit could not complete a block	P3v2/search_log_blockBS.json; audit/staged/repair12_receipt.json; PRECOMMIT_P3v2.md	3 (0)
10	the pre-registered report-flag rule counts the new clinic's name wherever a report mentions it, so silent-arm reports naming the distractor appointment's clinic were labelled as flagging the change (found by an archive audit of the model reports)	P3v2/FLAG_AUDIT.json; P3v2/search_log_blockB.json; PRECOMMIT_P3v2.md	3 (3)
11	the pre-registered rule voiding a re-exposure row whose panel frame was not delivered was never enforced, and delivery was redefined as text or image after the outcomes were known	P3v2/search_log_blockBS.json; PRECOMMIT_P3v2.md	3 (3)

X Extended related work

Section 2 positions this work against four lines; this appendix says what each shares with EnvOS and where it stops.

Benchmarks that make the environment move. OSWorld 2.0 (Yuan et al., 2026) is the closest in ambition: 31 services, long-horizon workflows, mid-task messages and a simulated user, and it reports the symptom this paper studies. Gaia2 (Froger et al., 2026), built in the ARE platform (Froger et al., 2025), lets asynchronous events change state while time flows, and pairs each scenario with a write-action verifier — the closest existing work to our invariant scoring. Real-Time Reasoning Gym (Wen et al., 2026) keeps the environment evolving while the agent is still reasoning and trades reasoning depth against latency; our change is one discrete intervention rather than continuous evolution of the environment. In τ^2-bench (Barres et al., 2025) a simulated user changes shared state through its own tools as part of a cooperative task, and InterruptBench (Zou et al., 2026) injects announced revisions of the user's goal into long-horizon WebArena-Lite tasks; our changes alter world state rather than the goal, and include silent arms with no message at all. DynamicGUIBench (Liu et al., 2026) changes interfaces between actions and answers with a video-input agent; AgentHijack (Sun et al., 2026) corrupts perception and control. StressWeb (Bai et al., 2026) compares perturbed with clean runs of the same tasks and finds agents that confidently report success on tasks they failed — the signature our silent arms also show — but its perturbations act on layout, interaction semantics and execution, not on authoritative backend state. STT-Arena (Hui et al., 2026) is the closest design: a trigger fires when a condition on state and spatio-temporal context holds, applies

Table 34: **Certification challenge.** For each of the 11 defects: the inference it permitted, the executable check that rejects it now, and the corrected conclusion; "rows (model)" counts archived search-log rows scored under the defect and how many already carried a model verdict.

#	inference the defect permitted	executable check that rejects it now	corrected conclusion	rows (model)
1	the clean control fails, so the family is hard	clean-control stop rule (`run_block`: a counted clean failure halts the block) + the memo channel declared by the task	harness lacked working memory; attempt voided; Block A2 clean 2/2	1 (1)
2	a session-limit reply is a model failure	provider-limit signal in the sealed invoker → `infra_rejected`, block pauses, reserve untouched	not counted; no failure rate moved	1 (1)
3	a changed-arm success with no change is evidence of recovery	exposure-conditioned trigger (`requires_seen`) + validity ladder branch `not_triggered` + pause rule	3 of 12 episodes were not instances; re-run as Block A2	11 (8)
4	correct runs read as unexposed	scorer places the change on the firing request; C4 recomputed; independent scorer cross-checks	exposure certified in 8/8 changed runs	3 (3)
5	the family's reference policy solves the silent arm, so its failures are model deficits	per-level certification of the expected matrix before any model: the ordinary `reference` must score 0.8 and the `late_reread_reference` 1.0	trigger timing kept (first wizard request after exposure); reference redefined to re-read the source inside the wizard (10/10 at 1.0), ordinary reference declared deficient; certificate 40/40	–
6	a 6-pair block is adequate for the rescue test	pre-registered power arithmetic (`envos.stats`) checked at freeze	6 unanimous pairs do reject ($p = 0.0312$), but one tie or one pair against forfeits rejection; enlarged to 10 pairs before any Block B rollout, which tolerates one pair against ($p = 0.021$)	–
7	a silent-arm stale booking is not a counted failure	ladder branch 5 skips `silent_arm` checkpoints; re-derived from archive	Block B row 1 re-derived from `delivery_ambiguous` to `model_failure`; OFF-A 0/10	–
8	the agent never dismissed the notice	independent scorer: raw access log vs C3, both directions	labels of 5 rows corrected; verdicts unchanged; dismissal 8/8	5 (5)
9	an errored read voids the episode	read-target rule voids only a successful read of a path the harness did not hand over	3 Sonnet attempts stayed void, cells replaced under the fixed rule; Block BS completed	3 (0)
10	a silent-arm run labelled flagged told the user about the change	archive audit re-reads every report and quotes the sentence behind each flag (`FLAG_AUDIT`)	labels kept; no silent-arm report states the move; §6 not met as registered for Block B	3 (3)
11	Sonnet's rescue-arm failures ignored a delivered panel	delivery standard applied to the re-exposure checkpoint; channel recorded per step	panel reached Sonnet as text only in all 3; confounded with delivery; Opus unaffected	3 (3)

an effect that can invalidate the agent's plan, and, for solvable tasks, is scored by rule-based checks on the final state; the benchmark is compared with non-dynamic environments in aggregate, and it names stale-state execution as a dominant failure mode. Its environments are Python tool APIs rather than a browser, its trigger conditions are defined on state and context rather than on what the agent has observed, and we found no per-instance paired test in it. Our design combines a change fired only after the agent has been served the value it supersedes, with that exposure certified from archives; inapplicable controls sharing the changed arm's trigger (and, for an announced change, its notice); and a matched re-exposure test. Mitigation has also been studied: Li & Shi (2026), discussed below, test observation-interface changes at benchmark scale. These benchmarks answer "do agents fail when the world moves"; the design here asks, under one particular change fired after one particular observation, whether one interface element—the review-page panel—removes the failure. The change itself is present in both arms, so its own causal effect is not estimated (in the historical blocks FRX-OFF-I was certified but never run on a model; Block BGC adds it). The two are complementary: breadth of setting against control of one mechanism.

Stale belief as an object of study. Reading a value and committing after it has been superseded is the classic time-of-check-to-time-of-use race. Lilienthal & Hong (2025) study it in LLM agents as a security vulnerability (TOCTOU-Bench, 66 tasks) and propose defences such as state-integrity monitoring; we measure it as a behavioural failure under a benign change fired inside the agent's own check-use window. Knowledge-conflict studies (Xie et al., 2024a) show that LLMs can adopt coherent contradicting evidence yet show confirmation bias toward what they already believe, in static question answering. STALE (Chao et al., 2026) asks whether a model notices that a stored memory has been invalidated, in question answering, without actions or a world to act on. Our silent

arms are the acting counterpart of both: the belief is formed by reading a page, the invalidation is a real row changing, and the cost is a durable write. EnvProbe (Song & Cai, 2026) approaches the same problem from the policy side, deciding when an agent should re-query a belief; our re-exposure arm is the environment-side counterpart. In the historical, hypothesis-generating clinic Block B (predecessor protocol; Appendix T), putting the current value on Opus 5's path immediately before the commit coincided with a 10/10 paired change in evaluator-recorded booked-clinic correctness, consistent with the interface carrying a burden the policy was not carrying; those successes are not verified on the full endpoint and are not confirmatory. The prospective claims blocks do not show this for Opus 5, which succeeded on 10/10 silent-change instances without the panel; there the panel contrast is Sonnet 5's (1/10 without, 9/10 with), for this task and interface. Li & Shi (2026) reach a compatible conclusion at benchmark scale: richer between-step observation (keyframes, audio transcription, narration) gains 17 to 48 points on 100 dynamic browser tasks, with a 50-task static control. Our re-exposure arm is narrower and pre-registered: one element, the current value on the agent's path immediately before the commit, tested on matched instances with an exact paired test.

Verifier validity and evaluation auditing. TheAgentCompany (Xu et al., 2025) self-hosts a reproducible workplace of services with simulated colleagues and checkpoint-based partial credit; our testbed is a workplace of self-hosted services in the same style. OpenComputer (Wei et al., 2026) builds verifiable software worlds with state verifiers and shows hard-coded verifiers agree with human adjudication better than a model judge, an empirical validation of outcome verifiers on static tasks. ABC (Zhu et al., 2025) turns task-setup and reward-design flaws found in existing agentic benchmarks into a checklist and shows such flaws can shift measured performance by up to 100% relative; our defect taxonomy is evidence of the same kind for one dynamic protocol, with protocol-specific checks for trigger timing relative to exposure and delivery of the evidence. Kirgis et al. (2026) argue that outcome-only agent evaluation is not credible and give a taxonomy of threats documented through execution-log analysis — the argument this paper partially executes, for the particular case of a dynamic intervention, where the verdict is not the only thing needing certification: the intervention, its timing relative to the agent's own observations, and the delivery of the evidence all do. Related critiques (Zhu et al., 2026; Patel et al., 2026) question what a single leaderboard number means at all. Our defect taxonomy (§6) is the empirical form of that argument: 7 of 11 defects had already produced rows carrying model verdicts.

Conventions we inherit. The testbed follows the task and scoring conventions of the computer-use, web-agent and tool-use benchmarks (Xie et al., 2024b; Zhou et al., 2024; Trivedi et al., 2024; Yao et al., 2025); the exact paired test is McNemar's (McNemar, 1947) and the intervals are Wilson's (Wilson, 1927). Neither statistical choice is novel here; what is pre-registered is which single contrast they are applied to.

Y MANIFESTS AND THE CLAIMS LEDGER

Release manifests. Block A2: 29 pinned repository paths, 2 container image ids, $\lambda = 0.5$, reward version 2, protocol version 2, manifest sha256 `b66756ff`. Block B: 29 pinned paths. A manifest pins the sha256 of every path it names, as hashed at freeze time; the rollout driver recomputes every one before each episode and refuses to start on any difference, which is what "the implementation did not change during the block" means here. The manifests are distributed as files; their path-by-path digest tables are not typeset.

Disposition of every registered block. Every block the pre-registrations name and what became of it (table below): completed (A2, B, BS; A2S and BF as the screens ADDENDUM 13 downsized them to), void or superseded (Block A attempt 1; the Block A restart), never started (D; P1MS), cancelled before any rollout (BR, BRS, by the consult decision ADDENDUM 9 deferred to; no addendum recorded it, as ADDENDUM 9 required), failed admission (BG, whose leak audit did not pass; P2S/P2SS, whose re-certification was rejected with P2-C 8; P2-L0 9; P2-L1 18 failed checks), and started but unevaluable (P1M). No outcome of any block that ran is omitted from this paper. The clean-control stop rule makes a stop a "harness finding" by rule, not by diagnosis: a valid clean task can fail by chance and would stop a block the same way. For Block A attempt 1, the one block it stopped, the attribution to the harness rests on ADDENDUM 2's separate diagnosis (the policy

had no working memory), not on the stop. Counts are read from the search logs; the category is the authors' reading of the cited artifact.

block	disposition	att.	cnt.	reason	artifact
A (attempt 1)	void	1	1	clean-control failure stopped the block (ADD. 2); harness lacked working memory	P3v2/_void_blockA_attempt1_2026-09-17/search_log_blockA.json
A (restart)	superseded	12	8	paused on the infrastructure rule; trigger replaced (ADD. 3); never pooled	P3v2/_superseded_blockA_2026-09-17/search_log_blockA.json
A2	completed	19	18	pre-registered primary test (Section U)	P3v2/BLOCK_A2_RESULT.json
A2S	completed (screen)	9	9	downsized, test withdrawn before any rollout (ADD. 13)	P3v2/BLOCK_A2S_RESULT.json
B	completed	23	23	pre-registered primary test (Appendix T)	P3v2/BLOCK_B_RESULT.json
BS	completed	26	23	one repair cycle (ADD. 12)	P3v2/BLOCK_BS_RESULT.json
BF	completed (screen)	5	5	downsized before any rollout (ADD. 13); silent arm and one control only	P3v2/BLOCK_BF_RESULT.json
D	never started	0	0	demoted to development, its rollouts released to Block B (ADD. 4)	jobs/continuation.json
BR	cancelled	0	0	consult round 13 skipped it before any rollout; ADD. 9 required an addendum saying so, none was written	jobs/continuation.json
BRS	cancelled	0	0	as BR	jobs/continuation.json
BG	failed admission	0	0	leak audit did not pass (the model's frame read failed); provider quota then exhausted	jobs2/world2.json
P1M	unevaluable	4	1	both intervention rollouts exhausted the budget before the committing request; paused	P1/search_log_blockP1M.json
P1MS	never started	0	0	blocked behind P1M	jobs2/world2.json
P2S	failed admission	0	0	re-certification rejected; admission gate never passed	jobs2/world2.json
P2SS	failed admission	0	0	as P2S	jobs2/world2.json

Claims ledger. *What it checks.* Every macro this paper uses is an entry of the ledger (`claims.json`, written by `claims.py`) that records its value, the artifacts it is derived from with their sha256, the statistical unit and the aggregation; `python claims.py -verify` re-derives every entry, fails on any drift of a pinned artifact and on any macro used but not defined, and checks the abstract's generated clauses, the main text's guarded wording and the interpretation templates' seal. *What is pinned.* Frozen artifacts are read in place against their digests (the claims blocks' evidence bundle against its digest list, `063b7ae44e11`); a live artifact (a search log its block was still writing; the scorer source) is read through a dated snapshot under `paper/snapshots/`, taken by the build, whose digest is the one pinned; an entry read with no snapshot has its drift reported, not failed, and the verification fails if the manuscript uses it, directly or inside another macro (in this build, no macro the manuscript uses is read from a live entry). *How to reproduce.* `./build.sh` snapshots the live artifacts, regenerates the ledger, every table and every figure, compiles the paper and runs the verification; `./build.sh -stress` builds a temporary copy with synthetic worst-case prospective results and checks its page limit. *What is missing.* Evidence that was never archived, or was overwritten, cannot be re-derived by any build; the next paragraph lists it. The per-macro ledger table (value, source artifact, digest prefix) is distributed with `claims.json` rather than typeset.

Supplement inventory and evidence gaps. *Records:* the frozen pre-registrations, the release manifests and the ledger (`claims.json`) of the paper build the package was made from ship as files in the supplement. The packaged supplement holds 24399 files (94.6 MB zipped). In one offline run of this package (no network, world or model), the claims target passed all 85 of its checks, every recomputed count, test, interval, reading and audit count equal to this ledger's entry, and the challenge target all 31, the re-run comparison's output identical to the shipped result; the packaged ledger's verification re-derived all 2319 of its entries. Its prospective target passed all 174 of its checks: Table 1's cells and Figure 3's per-instance grids for the claims blocks and Block BC, Block BC's paired tests, Holm adjustments, Newcombe intervals and audit counts, recomputed from the raw run directories, equal this ledger's entries; for Block BCS, which stopped unfinished at its registered cutoff and reports no test, its independent scorer and validity re-derivation

were re-run and equal their records, and its attempts, counted cells and per-instance table were re-computed. All 9 historical archive re-analysis re-derivations reproduced; every generated table (61 files) re-renders byte for byte from the shipped records. Its complete inventory—what is packaged, what exists but is not packaged and why, and what was not yet produced—is a generated supplementary file (`supp_provenance/`, Appendix U). *Permanently missing* (never archived, or overwritten; no package can supply them, and only new rollouts can replace what they would have shown): Block BS's first manifest (`9aa433099a46`), known by digest only; the two code versions pinned by the archived pre-image Block A manifest (the harness `e4e215d1` and the task module `1eeaab16`); the first outputs of Block A2's mutation battery and leak audit, overwritten later that day; the first transfer run's output; the scorer digest of the certification audit; the CLI's built-in system prompt, tool definitions, decoding and reasoning defaults and token usage; the results of non-`Read` tool calls, including whether Sonnet's 1591 other calls executed; per-step page titles and element lists, so the observation a policy received cannot be reconstructed exactly; server-side model identity; and slot occupancy at booking time and calendar compatibility in the historical blocks, so no historical success can be verified in full.

Z VIDEO INDEX

Videos are rendered from the archived frames of real episodes — no episode is re-run to make one — and each manifest records the frame digests and the source trajectory. The pairs that show the paper's result directly are `blockB_s20` and `blockB_s21`: the same instance under FRX-OFF-A and FRX-ON-A, where Claude Opus 5 commits the superseded clinic in the first and, in the second, goes back to change its selection after the review step re-renders the live row. `blockBS_s22` is the counterpart in which Sonnet 5 is *not* rescued. `blockA2_s8` and `blockA2_s14` show the announced arm, the second being the block's single failure.

file	duration (s)	sha256	episodes
ambiguous_commit_failure_demo.mp4	59.0	c3814b95 (trajectory)	claude-fable-5-1 AC-L4 changed None
blockA2_s14_control_vs_changed.mp4	198.04	0bdacf63 (video)	S-C S-test_id-14 APPLIED_CORRECTLY; S-L2 S-test_id-14 STALE_APPLIED_FLAGGED
blockA2_s8_control_vs_changed.mp4	208.04	182d1726 (video)	S-C S-test_id-8 APPLIED_CORRECTLY; S-L2 S-test_id-8 APPLIED_CORRECTLY
blockBS_s20_OFF-A_vs_ON-A_sonnet5.mp4	290.04	573303be (video)	FRX-OFF-A S-test_id-20 STALE_APPLIED_UNFLAGGED; FRX-ON-A S-test_id-20 REVISED_ON_REEXPOSURE
blockBS_s22_OFF-A_vs_ON-A_sonnet5_not_rescued.mp4	271.04	ebd34ddc (video)	FRX-OFF-A S-test_id-22 STALE_APPLIED_UNFLAGGED; FRX-ON-A S-test_id-22 STALE_APPLIED_FLAGGED
blockB_s20_OFF-A_vs_ON-A.mp4	203.04	12f8f746 (video)	FRX-OFF-A S-test_id-20 STALE_APPLIED_UNFLAGGED; FRX-ON-A S-test_id-20 REVISED_ON_REEXPOSURE
blockB_s21_OFF-A_vs_ON-A.mp4	239.04	9c2fa9a3 (video)	FRX-OFF-A S-test_id-21 STALE_APPLIED_UNFLAGGED; FRX-ON-A S-test_id-21 REVISED_ON_REEXPOSURE
p1_ambiguous_commit_AC-L3_fable_rlstyle.mp4	110.04	86ca27ea (video)	AC-L3 ambiguous_commit_booking ?
p3_named_target_s0_opus5_rlstyle.mp4	118.04	fe613332 (video)	P3-L2 superseded_clinic_after_ack STALE_APPLIED_UNFLAGGED
p3v2_blockA2_opus5_s14_failure.mp4	151.04	81823a6e (video)	S-C S-test_id-14 DELIVERY_AMBIGUOUS; S-L2 S-test_id-14 STALE_APPLIED_FLAGGED
p3v2_blockA2_opus5_s8.mp4	159.04	9254f31b (video)	S-C S-test_id-8 APPLIED_CORRECTLY; S-L2 S-test_id-8 APPLIED_CORRECTLY
p3v2_superseded_session_scripted_demo.mp4	157.04	7c4424ba (video)	S-L2 S-test_id-0 APPLIED_CORRECTLY; S-L2 S-test_id-0 STALE_APPLIED_UNFLAGGED
superseded_clinic_failure_demo.mp4	75.0	e61d438f (trajectory)	claude-fable-5-1 P3-L2 changed STALE_APPLIED_UNFLAGGED
superseded_clinic_failure_demo_opus5.mp4	75.0	b6766eb2 (trajectory)	claude-opus-5 P3-L2 changed STALE_APPLIED_UNFLAGGED