SEAM: REPRODUCIBILITY, AUDITABILITY, AND STABILITY FOR AGENTIC LLM SYSTEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Agentic systems need reproducible executions and evidence explaining why fresh decisions disagree. We present SEAM (Seal, Explain, Abstain, Mechanize), a client-side architecture that normalizes execution order, replays recorded interactions, tests explanations of divergence, withholds unsupported stability claims, and selects deterministic replacements. Controlled studies connect each operation to measurable outcomes. On a 100-slide CAMELYON17 lymph-node subset, each processed six times per arm, ordered admission restores bitwise agreement in 500 of 500 reference-run heatmap comparisons; a separate two-run analysis removes 203 matched-lesion size-class changes. Across four recordings of a model-directed ledger loop, ordering reduces cases with varying decisions from 4–5 of 6 to 0–1. A diagnostic panel with 30 cases and eight repeats exposes the cost of treating repeated calls as independent: the sealed arm's nominal one-sided 95% pair-risk upper bound is 0.068, while a finite-sample case-level bound for the same target is 0.262. On 300 three-factor attribution instances, sequential testing raises answer coverage from 51.0% to 85.3% with pruning held fixed; every returned sequential explanation matches the injected cause. A stricter procedure bounds erroneous returns under explicit sampling and separation assumptions. Greedy selection misses a beneficial joint replacement on one panel case. SEAM makes repeatability an execution property and stability claims accountable to recorded evidence.

1 INTRODUCTION

Agentic systems built around large language models (LLMs) choose tools, combine retrieved evidence and produce consequential decisions. When repeated executions disagree, developers need to reconstruct the execution, locate the source of divergence and assess whether a proposed change improves stability. Tracing and replay supply the execution record; repeated measurements and controlled interventions connect that record to an explanation and a remedy.

Variation originates on both sides of model and tool APIs. The client controls concurrent tool-result merges, budget admission and memory updates. Model providers control the serving numerics and available sampling settings (He & Thinking Machines Lab, 2025; Yuan et al., 2025). SEAM organizes these controls into an architecture for repeatability and auditability: normalize accessible sources of variation, preserve the evidence needed to reconstruct executions, and evaluate the variation that remains. We evaluate both fixed node graphs and model-directed loops.

The four components form a workflow (Figure 1). Seal records and normalizes execution; Explain tests candidate causes of divergence; Abstain withholds stability claims unsupported by the sampled evidence; Mechanize evaluates deterministic replacements. Appendix E details the implementation.

Contributions. We contribute an integrated architecture and controlled evaluations of its repeatability and audit mechanisms. Labels identify evidence by source: [A] for analytic results and model calculations, [M] for local experimental measurements, and [R] for billed-endpoint observations, including reanalysis of recorded evidence.

(1) *An execution and audit architecture.* SEAM connects ordering and replay with attribution, abstention and replacement selection. Image-pipeline and ledger-loop controls isolate what ordering resolves and what requires further analysis (§3–§4.2).

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.

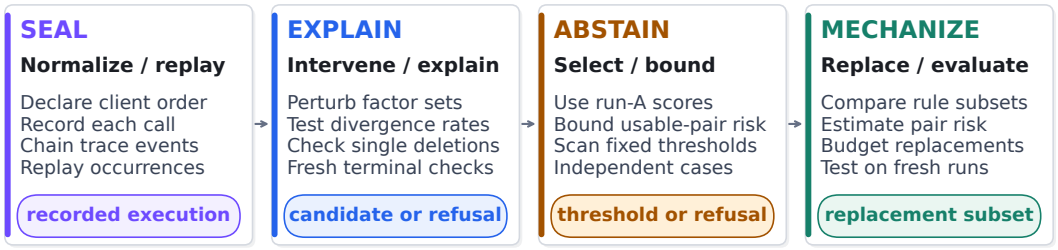

Figure 1: SEAM's workflow. Explain shows strict terminal checks; statistical claims require their stated sampling assumptions, and replacements require evaluation on fresh evidence.

(2) *Evidence requirements for stability claims.* A sampling audit shows how treating repeated calls as independent can overstate confidence. We distinguish pair disagreement from any disagreement across eight repeats (*any-of-eight risk*) to match the bound to the intended requirement (§4.3).

(3) *Tests of explanation and repair.* A pruning-matched ablation isolates sequential testing's contribution to cause recovery; terminal tests support a conditional minimality guarantee. Replacement experiments test whether component-level changes improve system-level agreement (§4.4, Appendix F.1).

Table 1 summarizes the designs and measurement units; Figure 2 in Appendix A.3 specifies the statistical procedures.

2 Preliminaries

2.1 Audit Targets and Units

A *case* is one workload input; a *repeat* is another execution of that case. A *trajectory* τ records model calls, tool calls and results in order. A *usable label* is a returned decision in the workload's allowed answer set. A *flip* is disagreement between two usable labels; missing or invalid answers are availability failures, reported separately.

Definition 1 (audit targets)**.** *Bitwise agreement* is equality of specified numerical output bytes. *Offline replay* re-executes with recorded model and tool responses in occurrence order. *Decision agreement* is equality of usable labels across fresh executions. *Attribution* identifies configuration factors whose perturbation retains an observed divergence, using interventions.

These are separate audit targets; none alone establishes decision correctness. A *certificate* is a statistical guarantee under explicit assumptions. *Coverage* is the probability that an audit procedure returns an answer. For score-based selection, it is the probability that a case's first execution passes the score rule (§3.3); for attribution, it is the probability of returning a factor set.

2.2 Related Work

Provider-side determinism. Batch-variant reductions and other numerical effects can change outputs even without sampling (He & Thinking Machines Lab, 2025; Yuan et al., 2025). Verified speculation offers a serving-stack intervention (Gond et al., 2026); client-side observations can reveal residual variation without removing its source (Fu et al., 2026). Changes over time pose a separate concern (Chen et al., 2024). Some APIs expose a best-effort seed and a system fingerprint (Anadkat, 2023); the endpoint used here exposes neither.

Delta debugging. Zeller & Hildebrandt (2002)'s `ddmin` procedure minimizes failure-inducing input by repeatedly deleting subsets and re-testing. Probabilistic monotonicity assessment accelerates minimization (Tao & Xue, 2025). Valle et al. (2026) apply delta debugging to cyber-physical systems with flaky tests, whose repeated executions can disagree (Luo et al., 2014). SEAM tests divergence-retaining subsets sequentially and evaluates that contribution with pruning held fixed (§4.4).

Selective risk control. Our threshold scan follows Learn-then-Test (Angelopoulos et al., 2025), using the risk–coverage formulation and binomial bound of selective classification (Geifman & El-Yaniv, 2017). Conformal risk control uses a different construction (Angelopoulos et al., 2024); subgroup and hierarchical methods address strata and grouped observations (Vovk et al., 2005; Dunn et al., 2023). Agent-specific work studies retrieval/tool-use drift and role-stratified risk (Opoku & Banahene, 2026; Rahman et al., 2026). We apply established testing to disagreement loss and audit its assumptions.

Repeated-run reliability. Sampled answers support aggregation (Wang et al., 2023), uncertainty estimation (Kuhn et al., 2023) and direct measurement of variation (Ouyang et al., 2025; Song et al., 2025; Atıl et al., 2025). Agent pass^k measures success on every repeated attempt (Yao et al., 2024). We use repeated-run disagreement to evaluate execution controls and the evidence supporting stability claims.

Deterministic replay. Instant Replay records shared-object access order (LeBlanc & Mellor-Crummey, 1987); ReVirt (Dunlap et al., 2002) and `rr` (O'Callahan et al., 2017) support whole-system and process-level replay, respectively. Content addressing and hash-chained logs provide established storage and integrity mechanisms (Merkle, 1988; Haber & Stornetta, 1991; Crosby & Wallach, 2009). Seal combines these primitives at agent tool and model boundaries, with explicit ordering and dependency declarations for repeated-run analysis (Appendix E).

Provenance and tracing. Workflow provenance (Moreau & Missier, 2013; Chirigati et al., 2016), distributed tracing (Sigelman et al., 2010) and OpenTelemetry conventions (OpenTelemetry Authors, 2026) supply established audit mechanisms. SEAM connects execution records to interventions and risk analysis. Our body-matched HTTP-cassette baseline matches SEAM's replay fidelity; step and dependency declarations supply additional information to SEAM's comparator (Appendix E).

3 SEAM

3.1 SEAL: A RECORD/REPLAY BOUNDARY WITH NORMALIZED CLIENT ORDER

Seal makes execution order explicit and preserves the interactions needed to reconstruct a run. The `SealedGraph` wrapper integrates with an agent's node graph (LangGraph in the reference implementation). A gateway intercepts model and tool calls, recording responses in a content-addressed store (CAS), keyed by response hashes. A request-hash index retains responses in occurrence order, so repeated identical requests can replay different recorded responses. Replay refuses missing or exhausted recordings and makes no external calls.

The wrapper fixes retrieval merge, budget admission and memory-commit order. Each step also declares the state fields it reads and writes, enabling trace comparison. Treating every field of the step's input state as a possible read is conservative only when that state includes every dependency. A file, clock or other external dependency must therefore be named in the step's read declaration or accessed through the recording gateway. A hash chain detects edits relative to a trusted digest; signed checkpoints additionally authenticate the expected terminal digest and length. Without a trusted checkpoint, the checker establishes internal consistency and leaves completeness unchecked.

§4.1 separates ordering from the audit layer: the gateway, store, chained trace and trace comparator. The comparator aligns traces by step index, identifies the first differing event hash (or the shorter length for a prefix), and returns the union of declared reads from that step onward. These are suspects for intervention testing: an earlier input can cause delayed divergence and escape this list.

3.2 EXPLAIN: STOCHASTIC DELTA DEBUGGING

Explain turns a trace difference into a testable account of what drives it. A *factor* is a named configuration dimension, such as tile arrival order. Stochastic delta debugging (SDD) searches for a small factor set whose perturbation retains the divergence. Because repeated interventions can yield different outcomes, a pilot sets a divergence-rate threshold and a sequential probability ratio test (SPRT) (Wald, 1945) classifies each tested subset. Sequential sampling spends more executions

on ambiguous subsets. Read-closure pruning first removes factors outside declared dependencies; `ddmin` searches the remaining subsets by repeated deletion.

The strict variant limits false explanations by capping the number of tests and refusing unresolved tests. Fresh terminal tests check that the candidate retains divergence and every single-member deletion falls below the threshold, including deletion of the only member. This property is *1-minimality*. Under fresh conditionally independent sampling with fixed subset probabilities, and separation of every queried probability from a specified ambiguity interval, Proposition 3 bounds the probability of returning a non-1-minimal set.

An acceptance gate also compares the candidate with size-matched random controls and single-member deletions to screen out weak explanations. Its empirical comparisons leave the theorem's sampling and separation assumptions to be justified independently. Historical SDD used threshold votes on unresolved tests and omitted fresh terminal checks; Appendix C distinguishes those results from strict mode.

3.3 ABSTAIN: SELECTIVE PAIR-RISK CONTROL

Abstain determines which executions have enough evidence to support a stated disagreement-risk target ε. For a case X in a predeclared subgroup (*stratum*) g, consider executions A and B independent conditional on X. Let U mean both return usable labels Y_A, Y_B. An instability score $s(X, A)$ uses A alone. The rule selects A when it is usable and its score is at most λ, targeting *selective pair risk*

$$R_g(\lambda) = \Pr(Y_A \neq Y_B \mid G = g,\ s(X, A) \leq \lambda,\ U).$$

This population average describes selected usable pairs. Coverage measures selection of A; failures of B are reported separately as availability failures. To choose λ, scan a list of candidate thresholds fixed independently of calibration data. Stop at the first upper risk bound exceeding ε, or an empty selected usable sample, and return the last passing threshold. Refuse if none passes.

Proposition 1 (threshold validity)**.** *For an independently fixed ordered list of thresholds, each with a valid $1 - \delta$ upper risk bound, the scan returns a threshold with risk above ε with probability at most δ. It can reach any such threshold only by passing the first unsafe threshold in the list, whose bound fails with probability at most δ (Appendix A.1).*

For binary disagreement, a Clopper–Pearson (CP) bound supplies the required upper bound when losses are independent and identically distributed (iid). The certificate-eligible path therefore uses one predeclared pair per independent case, with an independently fixed threshold list and run-A-only score. Recording these choices makes the assumptions auditable; independence still depends on how cases and executions were sampled. Repeated pairs within a case instead require cluster-level inference (§4.3). Confidence is per stratum; simultaneous coverage of all strata requires an additional correction. Appendix G separates historical calibration diagnostics from this scan.

3.4 MECHANIZE: SELECT DETERMINISTIC REPLACEMENTS

Mechanize asks which model-driven decision components to replace with deterministic rules under a budget. We call each available replacement a *verifier*. The objective is pair-disagreement risk, $\mathrm{PDR} = \Pr(Y_A \neq Y_B \mid U)$, over usable pairs. Component-level determinism alone is insufficient: Proposition 2 constructs an OR decision whose PDR increases after one component is replaced by its correct value. Replacement selection must therefore evaluate the resulting system-level decision.

Mechanize estimates disagreement for candidate verifier subsets. Its greedy rule buys a replacement only when the estimated PDR strictly decreases. Estimation error can reverse the realized effect, and interacting replacements can improve agreement even when neither helps alone. Full mechanization buys all available verifiers. The diagnostic-panel experiment compares greedy and joint selection, with case-level uncertainty and replication across recordings (§4.3; Appendix F.1).

4 EXPERIMENTS

Results come from recorded outcomes or declared simulations; the anonymous supplement includes their analysis code and recorded evidence. Billed campaigns send no sampling parameters. All

three tested model IDs reject `temperature`; the loop's model also rejects `top_k` and `top_p`. The API documents no seed. Default responses include reasoning blocks within the 256- or 1,024-token output cap; a separate loop experiment disables reasoning (Appendices B and D.7).

Table 1: Experimental design. Repeats improve measurement within a case; independent-case counts determine the scope of case-level inference.

study	cases / repeats	intervention and control	measurement / unit
images [M]	20 and 100 slides / 6	seeded orders; ordered-only; ordered + audit	bitwise agreement, size-class changes / slide
banking [R]	12 tickets / 20	raced vs ordered retrieval	usable-label disagreement, availability / ticket
panel [R]	30 and 90 cases / 8	raced, ranked fixed, pinned windows	pair risk; any-of-eight risk / case
attribution [M]	300 instances per tier and seed	fixed vs sequential tests $\times$ pruning off/on	cause recovery, 1-minimality, cost / instance
loop [R,M]	6 cases / 12	raced vs ordered; fixed selector; local decoding	tool-sequence and decision variation / case
replacement [R,M]	30 cases / 8	two verifier sets; all role subsets; greedy vs joint	PDR and gold agreement / case

4.1 Image Pipeline: Ordering versus Audit Instrumentation [M]

Setup. We process CAMELYON17 whole-slide images of hematoxylin-and-eosin-stained sentinel lymph nodes (Litjens et al., 2018; Bandi et al., 2019): initially 20 slides from 4 challenge patient groups, each processed six times per arm at pyramid level 4, the fifth stored resolution with level 0 at full resolution (Appendix B). Lesion size classes are isolated tumor cells (< 0.2 mm), micrometastasis (0.2–2.0 mm), and macrometastasis (> 2.0 mm), used in pathological nodal (pN) staging. A 101,793-parameter convolutional neural network (CNN) was trained on 460 patches from two slides of one patient. Both training slides and the 112-patch holdout slide belong to the evaluated cohort. We use its cached scores to isolate the effects of execution order: seeded permutations change tile admission and accumulation while detector outputs stay fixed. This design measures sensitivity to scheduler variation.

Two mechanisms. With detector outputs fixed, permuting accumulation order changes a tile probability by one to two float32 units in the last place (ULP), 5.96×10^{-8} to 1.19×10^{-7} (Appendix D.2). The much larger observed $\max|\Delta P| = 0.99998$ arises from admission: the probed slide (patient 017, node 2) offers 7,679 tiles against a 6,000-tile budget. Filling it in arrival order drops 1,679 tiles and lets runs score different slide regions. The budget was chosen to bind and does so on 11 of 20 slides. Declared raster order removes this variation; declared score order also achieves bitwise agreement (100/100, 0 flips) while admitting tiles with 22% higher mean detector scores (Appendix D.1). Thus reproducibility does not require raster order specifically. Figure 3 traces admission differences through the image pipeline.

Ordering versus explanation. Ordering alone recovers bitwise agreement and removes all 29 lesion size-class flips; the audit layer adds replay and trace comparison (Table 2). At finer declaration granularity (144 steps, 231 reads), the first differing step is heatmap accumulation; with accumulation pinned, it is admission. The comparator still returns 227 and 209 of 231 reads, respectively. Read-closure pruning and `ddmin`, not the comparator alone, narrow 39 suspect factors to 5 and then 1. Historical SDD selects patient 017 node 2's arrival order in three search tests, without establishing the strict theorem's assumptions. Independent interventions confirm its effect: 100 of 100 pairs disagree with all factors perturbed, 0 of 100 with that factor held (one-sided 95% upper bound 0.030), 100 of 100 with each of three controls held, and 0 of 100 on every slide after removing the budget (Appendix C.2).

Where the cascade stops, and why. pN flips are 0 in every arm. Although node labels change (patient 024 node 0 switches micro/macro in 20 of 100 all-perturbed pairs), every staged patient is assigned pN2 with five positive nodes, so no single node-label change alters the stage (Appendix C.2). The nominal one-sided 95% binomial upper bound 0.181 uses 15 consecutive-run comparisons from

Table 2: Original 20-slide ablation. Heatmap/tile denominators are 20 slides × 5 comparisons with run 0; lesion-class changes use a separate two-run cascade. All arms use detector `472c3e06...`. Localization uses one step per slide (coarse) or 144 steps with 231 reads (fine; Appendix C.2).

	A0 unsealed	**A1 ordered only**	**A2 ordered + SEAM**
heatmaps: bitwise agreement	0/100	100/100	100/100
tiles identical	45/100	100/100	100/100
lesion size-class flips	29	0	0
cross-run replay	miss	n/a	ok
comparator: runs identical	false	n/a	true
first divergent step named	0 (coarse); 4 (fine)	n/a	none
suspect read-set (union rule)	all 61 (coarse); 227 of 231 (fine)	n/a	no divergence

only three groups, limiting its interpretation. Stable stages are nevertheless inaccurate: exact stage agreement is 0 of 3 in every arm, and every arm's lesion set hits only 1 of the 131 annotated components on the 9 annotated slides. Ordering stabilizes the detector's outputs while leaving this localization failure unchanged. Across all detected lesions, the median distance to a size-class threshold is 0.23 mm (Appendix D.1).

Expanded validation. A frozen 100-slide extension retains the original 20 slides and balances 20 challenge groups across five centers. The same detector and level-4 protocol reproduce the original baseline in an updated execution environment. Both ordered arms then achieve 500/500 heatmap matches versus 0/500 unsealed; the separate two-run cascade yields 203 unsealed class changes and zero ordered, including 174 changes on the added 80 slides (Appendix D.4).

4.2 Ledger Loop: Residual Variation after Ordering [R]

The image pipeline has a predetermined node schedule (a *fixed graph*). In contrast, this agent reconciles an open amount against a sharded ledger by choosing its next query at run time. Six in-process shards have simulated latency; arrival-order selection and timing-dependent retries create client-side variation. The sealed arm fixes these orders but leaves model choices live. Across four recordings (*draws*), 576 episodes require 2,056 billed model calls on `anthropic.claude-sonnet-5`, of which 4 return unusable responses (Appendix B). A deterministic rule on retrieved results decides whether to settle, escalate or refuse the transaction. All three count as usable decisions; audit refusal instead means withholding an explanation or stability claim.

We measure the *tool sequence*: ordered `search_ledger(query,attempt)` strings, including retries but excluding model calls and results. Its length L_{tool} counts tool attempts. Comparing this sequence and the final label separately distinguishes variation in tool use from variation in decisions.

Table 3: Ledger loop: 6 cases × 12 repeats per arm in each of four billed recordings. Lists follow recording IDs 1–4; the replay row includes recording 5. L_{tool} counts at most 24 tool attempts. Each sealed recording starts 72 episodes; 1, 2, 1 and 0 fail to complete because of unusable model responses. Replay checks tool-sequence and decision equality separately (Appendix D.5).

	unsealed	**ordered + SEAM**
cases with byte-identical first model input	0 / 6 in each	6 / 6 in each
cases whose L_{tool} varied (of 6)	6, 6, 6, 6	2, 4, 3, 4
distinct tool sequences / episodes completed	69/72, 71/72, 69/72, 69/72	12/71, 16/70, 14/71, 18/72
cases whose tool sequence varied (of 6)	6, 6, 6, 6	5, 6, 6, 6
cases whose decision varied (of 6)	4, 5, 5, 4	1, 1, 0, 0
tool-sequence replay; internal trace check	n/a	equal in 4 of 5 (recordings 1–3 and 5)

Identical initial inputs, varying control flow. The seal holds the first model input byte-identical in all four recordings, while tool sequences vary in 5 or 6 of the 6 cases. A provider-side choice at step k changes the state composing the prompt at step $k + 1$: we call this *output propagation*. A fixed graph can propagate the same variation (§4.3); the loop additionally lets it change subsequent control flow and execution length.

Controls identify a sampling contribution. With a fixed selector replacing the model, every sealed case has one tool sequence in all 30 local draws (Appendix D.5). Locally served Qwen2.5-14B-Instruct produces six tool sequences over 72 sealed episodes in each greedy-decoding recording. Across both arms of the first recording, all 1,481 pairs of calls sharing identical prompts agree on their normalized next-step choice. Enabling sampling yields 18 tool sequences and within-prompt choice disagreement 0.161, isolating a sampling contribution within that setup.

At the provider endpoint, identical-prompt disagreement is 0.084–0.143 across the four default recordings and 0.022 in a reasoning-disabled recording, which also has no empty answers. This comparison changes both execution configuration and downstream prompts, leaving sampling and serving numerics unresolved as separate causes. Appendix B documents replay outcomes and failure instrumentation.

4.3 Pair Risk, Sampling Assumptions, and Repeat-Set Risk [R]

The Abstain component needs a sampling unit and a risk target before it can support a stability claim. We examine these choices using two billed campaigns (Appendix G). Retrieval races use in-process threads with unseeded latency jitter. Counts describe the recorded benchmark cases; population-bound calculations assess what an iid-case model with a fixed execution policy would support. The benchmark selection schemes and provider changes limit transfer to a deployment population (Appendix B.1).

Campaign A: banking77 intent triage (Casanueva et al., 2020). Each of four recordings uses 12 tickets $\times$ 20 repeats $\times$ 2 arms = 480 calls on `anthropic.claude-fable-5`. Every ticket's usable labels agree across repeats in both arms. Two 256-token recordings contain respectively 6 and 5 unusable answers out of 240 unsealed calls, versus 0 sealed. For the 6-versus-0 contrast, a Fisher exact one-sided test gives nominal $p = 0.015$ (Fisher, 1922), treating calls as independent despite clustering by ticket. Replications at 1,024 and 256 tokens each have 0 of 480 unusable answers, so the availability difference does not persist. Input normalization does persist: sealing produces 12 distinct prompts for 12 tickets, versus 103–112 with raced retrieval (Appendix G).

Campaign B: a three-agent diagnostic panel. Cases are sampled for near-equal diagnosis representation. Two LLM agents prompted as diagnostic specialists vote on each synthetic case. A third, prompted as a critic, receives their provisional diagnosis (the *leader*); a fixed voting rule produces the final label. The study uses 30 cases $\times$ 8 repeats $\times$ 3 calls $\times$ 2 arms = 1,440 calls on `anthropic.claude-opus-5`, with 0 failures. Sealing reduces distinct retrieved-evidence windows from 212 to the floor of 30, but distinct model inputs from 638 to 92, above the floor of 90. In 2 cases, varying specialist votes on identical prompts change the leader and critic prompt. This output propagation recurs in 9 of 90 cases in the larger study. Same-prompt pair disagreement is 3.9% for specialists and 7.8% for the critic.

A *pinned* control fixes each case's window to its unsealed first-repeat content (720 calls). Both fixed-window policies have low observed disagreement (Table 4). Their comparison remains sensitive to content and recording date because arms were not randomly interleaved (Appendix G).

Account for repeated measurements. Consecutive pairs reuse executions; disjoint pairs still cluster within cases. For the same equal-case-average pair risk, one-sided 95% Hoeffding bounds on independent case means are 0.262 and 0.257 in the sealed arm, above $\varepsilon = 0.1$. Approximate case-bootstrap bounds are 0.076 and 0.067. At 90 cases the Hoeffding bounds remain 0.175 and 0.173 (Appendix G). The finite-sample cluster analysis therefore withholds a pair-risk certificate despite the nominal pooled bounds below 0.1.

Match the deployment requirement. Requiring agreement throughout eight executions changes the target. With usable labels conditionally iid given X and label probabilities $p_l(X)$, pair risk is $\mathbb{E}_X[1 - \sum_l p_l(X)^2]$; any-of-eight risk is $\mathbb{E}_X[1 - \sum_l p_l(X)^8]$. The sealed any-of-eight upper bound is 0.2386 at 30 cases and 0.2071 at 90 (12 events). The 90-case unsealed bound is 0.2324 for 14

Table 4: Campaign B, 30 cases × 8 repeats. Parentheses show one-sided 95% Clopper–Pearson upper bounds (Clopper & Pearson, 1934). Pair bounds are nominal because repeated pairs cluster within cases. The any-of-eight row counts cases with disagreement across eight repeats; its binomial bound requires iid cases.

event counted	unsealed	sealed	pinned	interpretation
consecutive-pair flip	25/210 (0.1623)	8/210 (0.0677)	6/210 (0.0556)	nominal pair bound
disjoint-pair flip	16/120 (0.1954)	4/120 (0.0747)	3/120 (0.0633)	nominal pair bound
any-of-eight event	9/30 (0.4651)	3/30 (0.2386)	2/30 (0.1953)	separate target

observed-repeat events, with one case lacking its eighth repeat and hence only partially observed. These bounds exceed 0.1; comparing the arms' risks additionally requires a between-arm analysis.

Strata and sample size. With zero any-of-eight events, the iid-case upper bound is $1-\delta^{1/n}$: 0.1391, 0.3930 and 0.5271 for the 20-, 6- and 4-case strata. At 95% confidence, at least 29 independent cases per stratum are needed even with zero events. In the larger campaign the afebrile (no-fever) stratum has 51 cases and four events, upper bound 0.1706; additional independent observations could certify a true risk below 0.1. In the hypoxic (low-oxygen) stratum, two cases flip under sealed inputs and none under pinned inputs; content, case identity and recording time remain confounded. Appendix G gives the strata and retains historical pooled-pair threshold calculations as nominal diagnostics.

Deterministic replacement on this panel. A signature-based verifier fitted on the 4,970 corpus cases outside this sample replaces the critic on the same recorded specialist outputs. Its plug-in pair risk changes from 0.0354 to 0.0292; the paired case-bootstrap difference is -0.0063 with 95% interval $[-0.0375, 0.0188]$. Agreement with the dataset label (*gold agreement*) changes by 0.0083, driven by one case. The disagreement change reverses on the pinned recording ($+0.0208$), so neither a robust stability gain nor accuracy non-inferiority is established. Two identical verifier votes determine the decision and yield zero disagreement by construction. Greedy allocation misses such a joint replacement on one case where no single replacement strictly reduces estimated pair-disagreement risk. Appendix F.1 reports both verifier sets, all subsets and uncertainty.

4.4 Attribution: Recovery, Refusal, and Matched Testing [M]

Design and metrics. DRIFT-Bench generates attribution problems in five difficulty tiers, D1–D5, with known injected causes (Table 10). D1 triggers on one cause factor, D2 on either of two, D3 on both of two, and D4–D5 on all three. D5 adds high background divergence and a weak injected effect. Coverage is the fraction of instances for which SDD returns a nonempty factor set. *Cause-family exact match* means the returned set is an injected minimal cause: in D2, returning either of the two cause factors alone is correct. *Threshold 1-minimality* means the set retains divergence above the pilot threshold and every single deletion falls below it. Background divergence and threshold choice can make this differ from recovering the injected cause (Appendix C).

Matched comparison. To separate sequential testing's effect from search-space reduction, we cross fixed-repeat versus sequential tests with pruning off versus on. Arms share instances, pilot/pruning draws and the acceptance gate over the full factor universe; separate random-number streams keep these draws matched. Fixed-repeat counts were calibrated at $\alpha = \beta = 0.02$ and reused at $\alpha = \beta = 0.1/(334 + 0.1)$, so total execution costs differ. At the first setting, sequential testing raises D3 coverage from 0.893 to 0.993 without pruning and from 0.890 to 0.993 with pruning. D4 coverage rises from 0.510 to 0.850 and 0.853, respectively. All returned sequential sets exactly match the injected cause family on both tiers. Sequential testing thus recovers more causes under either pruning policy. Table 5 reports both calibrations and costs; Table 9 retains the historical five-seed results.

Guarantee versus observation. With a predeclared cap of 334 tests, the bound at $\alpha = \beta = 0.02$ exceeds 1 on every tier and gives no useful probability guarantee. Strict SDD instead uses $\alpha = \beta = 0.1/(334 + 0.1)$ for a bound of 0.1, conditional on endpoint separation and fresh sampling. In the matched strict supplement, pruning-on coverage is 1.000, 0.640 and 0.0133 on D3–D5, with mean total executions 1,870, 2,828 and 848. No returned answer has an observed cause-family or

threshold-minimality error. Separation remains an assumption, especially on D5, where early refusal explains the lower cost.

D5 also tests whether refusal reflects the attribution procedure's limits. A contrast test that compares each candidate with controls from the same factor layers recovers the injected cause in 196/300 instances, with every returned set exact. Its cost is 24,050 executions per instance, roughly seven times historical SDD's cost (Appendix C.1). D5 is therefore recoverable with a different test at substantially greater cost; Appendix C separates cause recovery from the historical scoring policy that rewarded refusal on this tier.

Table 5: Matched membership-test ablation, seed 4, 300 instances per tier. Cells are coverage / mean total executions, including pilot, pruning and gate. Fixed-k uses a calibrated repeat count; total costs differ. "Shipped" and "corrected" denote the two error settings in the text. These arms use threshold-vote fallback at timeout; strict results are reported separately.

calibration	**pruning**	**tier**	**fixed-k**	**sequential**
shipped	off	D3	.893 / 1196	.993 / 1101
shipped	on	D3	.890 / 1187	.993 / 1095
shipped	off	D4	.510 / 2633	.850 / 2707
shipped	on	D4	.510 / 2635	.853 / 2733
corrected	off	D3	.893 / 1196	1.000 / 1822
corrected	on	D3	.890 / 1187	1.000 / 1803
corrected	off	D4	.510 / 2633	.943 / 4599
corrected	on	D4	.510 / 2635	.947 / 4623

5 LIMITATIONS

Scope. The experiments isolate SEAM's mechanisms in limited workloads. The image pipeline has no language model and uses a weak detector; the loop has six constructed cases and a simulated race; the synthetic panel has low gold agreement. Generalization requires more diverse cases, failures and providers. Attribution is tested mostly on our own generator, with one real-data factor confirmed by intervention.

Guarantees. Selective risk control requires an independently fixed threshold list and valid sampling-based bounds. Strict SDD additionally requires endpoint separation and fresh conditional sampling. Verifier allocation optimizes estimated disagreement; population safety requires separate evaluation.

Audit comparison. Measured loop overhead is 0.66 ms and 780 bytes per tool step (Appendix B). The body-matched cassette plus hash chain matches SEAM on replay of recorded calls, refusal of diverging requests and detection of tested single-bit record mutations. It lacks the step declarations needed by SEAM's comparator, but a declaration-aware event log could supply them. The current union-of-reads comparator barely narrows the real divergence's suspects; comparative debugging cost remains unmeasured. Trusted anchors are required for adversarial integrity claims.

6 CONCLUSION

SEAM brings execution control and residual-variation analysis into one audit architecture. The image-pipeline and ledger studies show why this combination matters: ordering removes a concrete client-side source of disagreement, while fresh model choices expose the limits of replay as evidence of stability. Sequential attribution provides a way to investigate that remaining variation, with the matched experiments demonstrating improved cause recovery. The diagnostic panel shows why claims of stability must account for case dependence and why repairs must be judged by their joint effect on the final decision. SEAM makes reproducibility a basis for justified changes to agent behavior.

Ethics Statement

This study involves no human participants or protected health information. CAMELYON17 is public under CC0 from the Computational Pathology Group at Radboud University Medical Center. Banking77's original release is CC-BY-4.0, while the MTEB mirror's metadata lists MIT. We retain attribution to the original release (Casanueva et al., 2020). Campaign B uses pre-existing synthetic corpora under CC0-1.0 and MIT, with source licenses recorded in `sources.jsonl`. Their labels are synthetic and lack clinical validation. The software is a research instrument and is not for diagnostic use: the image detector has train/test overlap and poor diagnostic agreement, and the panel is evaluated against synthetic dataset labels. Reproducibility must therefore be assessed separately from clinical utility.

Reproducibility Statement

Appendix A gives the assumptions and proofs for Propositions 1–3. Appendix B maps results to analysis scripts and recordings, documents the environment, seeds and detector digest, and specifies regeneration and replay limits. Appendix C provides attribution settings and cost accounting. The anonymous supplement includes the recorded evidence from billed campaigns, cited analysis code, manuscript sources and drift tests, with a file-checksum manifest and offline validation instructions. The campaign tables can be recomputed from this evidence without credentials or provider calls. Fresh live execution requires explicit spending approval and enforced call caps; the library's live entry point additionally requires a signed budget.

AI Use Statement

Mathematical claims and proofs. AI assisted in drafting and reviewing Propositions 1–3. Appendix A states their arguments; tests check numerical counterexamples and implementation behavior.

Methodology and experiments. AI-assisted review identified problems in sampling assumptions, attribution calibration, and comparison design. AI also assisted with analysis code and experimental harness implementation. Quantitative results are computed from recorded evidence or explicitly labeled simulations.

Data and references. Clinical cases are seeded samples from pre-existing licensed synthetic datasets, selected and normalized by code. We did not use generative AI to create or clean dataset content. AI assisted with literature-search and reference-formatting suggestions; bibliographic details were checked against primary records.

Writing and figures. AI assisted with prose, LaTeX and figure-producing code. Data-bearing figures are computed from recorded evidence or declared simulations. The authors remain responsible for reviewing the code, evidence, proofs, interpretations and final manuscript.

References

Shyamal Anadkat. How to make your completions outputs consistent with the new seed parameter. OpenAI Cookbook. `https://developers.openai.com/cookbook/examples/reproducible_outputs_with_the_seed_parameter`, 2023. Accessed 2026-09-05.

Anastasios N. Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. Conformal risk control. In *International Conference on Learning Representations (ICLR)*, 2024. URL `https://openreview.net/forum?id=33XGfHLtZg`.

Anastasios N. Angelopoulos, Stephen Bates, Emmanuel J. Candès, Michael I. Jordan, and Lihua Lei. Learn then test: Calibrating predictive algorithms to achieve risk control. *The Annals of Applied Statistics*, 19(2), 2025. doi: 10.1214/24-AOAS1998.

Berk Atıl, Sarp Aykent, Alexa Chittams, Lisheng Fu, Rebecca J. Passonneau, Evan Radcliffe, Guru Rajan Rajagopal, Adam Sloan, Tomasz Tudrej, Ferhan Ture, Zhe Wu, Lixinyu Xu, and Breck Baldwin. Non-determinism of "deterministic" LLM system settings in hosted environments. In *Proceedings of the 5th Workshop on Evaluation and Comparison of NLP Systems*, pp. 135–148, 2025. doi: 10.18653/v1/2025.eval4nlp-1.12.

AWS Open Data Registry. CAMELYON — registry of open data on AWS. https://registry.opendata.aws/camelyon/, n.d. Accessed 2026-09-11.

Peter Bandi, Oscar Geessink, Quirine Manson, et al. From detection of individual metastases to classification of lymph node status at the patient level: The CAMELYON17 challenge. *IEEE Transactions on Medical Imaging*, 38(2):550–560, 2019. doi: 10.1109/TMI.2018.2867350. PMID: 30716025.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI (NLP4ConvAI)*, pp. 38–45, 2020. doi: 10.18653/v1/2020.nlp4convai-1.5.

Lingjiao Chen, Matei Zaharia, and James Zou. How is ChatGPT's behavior changing over time? *Harvard Data Science Review*, 6(2), 2024. doi: 10.1162/99608f92.5317da47.

Fernando Chirigati, Rémi Rampin, Dennis Shasha, and Juliana Freire. ReproZip: Computational reproducibility with ease. In *Proceedings of the 2016 International Conference on Management of Data*, pp. 2085–2088, 2016. doi: 10.1145/2882903.2899401.

C. J. Clopper and E. S. Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 1934. doi: 10.1093/biomet/26.4.404.

Scott A. Crosby and Dan S. Wallach. Efficient data structures for tamper-evident logging. In *Proceedings of the 18th USENIX Security Symposium*, pp. 317–334, 2009. URL https://static.usenix.org/event/sec09/tech/full_papers/crosby.pdf.

George W. Dunlap, Samuel T. King, Sukru Cinar, Murtaza A. Basrai, and Peter M. Chen. ReVirt: Enabling intrusion analysis through virtual-machine logging and replay. *ACM SIGOPS Operating Systems Review*, 36(SI):211–224, 2002. doi: 10.1145/844128.844148.

Robin Dunn, Larry Wasserman, and Aaditya Ramdas. Distribution-free prediction sets for two-layer hierarchical models. *Journal of the American Statistical Association*, 118(544):2491–2502, 2023. doi: 10.1080/01621459.2022.2060112.

R. A. Fisher. On the interpretation of χ^2 from contingency tables, and the calculation of P. *Journal of the Royal Statistical Society*, 85(1):87–94, 1922. doi: 10.2307/2340521.

Tairan Fu, Gonzalo Martínez, Javier Conde, Carlos Arriaga, Xiuyuan Qi, Shanshan Liu, and Pedro Reviriego. Beyond reproducibility: Token probabilities expose large language model nondeterminism. *IEEE Transactions on Computers*, pp. 1–14, 2026. doi: 10.1109/TC.2026.3732142.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems*, volume 30, pp. 4878–4887, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/4a8423d5e91fda00bb7e46540e2b0cf1-Abstract.html.

Raja Gond, Aditya K. Kamath, Ramachandran Ramjee, and Ashish Panwar. LLM-42: Enabling determinism in LLM inference with verified speculation. *arXiv preprint arXiv:2601.17768*, 2026. URL https://arxiv.org/abs/2601.17768.

Stuart Haber and W. Scott Stornetta. How to time-stamp a digital document. *Journal of Cryptology*, 3(2):99–111, 1991. doi: 10.1007/BF00196791.

Horace He and Thinking Machines Lab. Defeating nondeterminism in LLM inference. Thinking Machines Lab: Connectionism, 2025. URL https://thinkingmachines.ai/blog/defeating-nondeterminism-in-llm-inference/. doi: 10.64434/tml.20250910.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *International Conference on Learning Representations (ICLR)*, 2023.

Thomas J. LeBlanc and John M. Mellor-Crummey. Debugging parallel programs with instant replay. *IEEE Transactions on Computers*, C-36(4):471–482, 1987. doi: 10.1109/TC.1987.1676929.

Geert Litjens, Peter Bandi, Babak Ehteshami Bejnordi, et al. 1399 H&E-stained sentinel lymph node sections of breast cancer patients: the CAMELYON dataset. *GigaScience*, 7(6):giy065, 2018. doi: 10.1093/gigascience/giy065. PMID: 29860392.

Qingzhou Luo, Farah Hariri, Lamyaa Eloussi, and Darko Marinov. An empirical analysis of flaky tests. In *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE)*, pp. 643–653, 2014. doi: 10.1145/2635868.2635920.

Ralph C. Merkle. A digital signature based on a conventional encryption function. In *Advances in Cryptology — CRYPTO '87*, Lecture Notes in Computer Science, pp. 369–378, 1988. doi: 10.1007/3-540-48184-2_32.

Luc Moreau and Paolo Missier. PROV-DM: The PROV data model. W3C recommendation, World Wide Web Consortium, April 2013. URL https://www.w3.org/TR/2013/REC-prov-dm-20130430/. Edited by Luc Moreau and Paolo Missier. Published 30 April 2013.

mteb. mteb/banking77. Hugging Face Datasets, revision 18072d2685ea682290f7b8924d94c62acc19c0b2. https://huggingface.co/datasets/mteb/banking77/tree/18072d2685ea682290f7b8924d94c62acc19c0b2, 2025. Parquet test split: 3,076 rows; legacy test.jsonl: 3,080 rows. Dataset card declares MIT; derived from the Banking77 release of Casanueva et al. (2020), distributed by PolyAI under CC-BY-4.0.

Robert O'Callahan, Chris Jones, Nathan Froyd, Kyle Huey, Albert Noll, and Nimrod Partush. Engineering record and replay for deployability: Extended technical report. *arXiv preprint arXiv:1705.05937*, 2017. URL https://arxiv.org/abs/1705.05937.

OpenTelemetry Authors. OpenTelemetry GenAI semantic conventions. https://github.com/open-telemetry/semantic-conventions-genai, 2026. Accessed 2026-09-05.

Jeffery Opoku and David Banahene. ToolChain-CRC: Conformal risk control for agentic AI under retrieval and tool-use drift. *arXiv preprint arXiv:2606.18467*, 2026. URL https://arxiv.org/abs/2606.18467.

Shuyin Ouyang, Jie M. Zhang, Mark Harman, and Meng Wang. An empirical study of the non-determinism of ChatGPT in code generation. *ACM Transactions on Software Engineering and Methodology*, 34(2):1–28, 2025. doi: 10.1145/3697010.

Md Ashikur Rahman, Md Arifur Rahman, Niamul Hassan Samin, Khandaker Rifah Tasnia, Md Hasibul Amin, Sifat Rahman Ahona, and Juena Ahmed Noshin. Beyond aggregate risk: Role-stratified conformal risk control for LLM tool calls. *arXiv preprint arXiv:2607.24343*, 2026. URL https://arxiv.org/abs/2607.24343.

Mahsa Sanaei. Disease symptoms and treatments dataset. Kaggle, MIT license. https://www.kaggle.com/datasets/snmahsa/disease-symptoms-and-treatments-dataset, 2025. 405 rows, used only for exact diagnosis-to-treatment matches; accessed 2026-08-07.

Benjamin H. Sigelman, Luiz André Barroso, Mike Burrows, Pat Stephenson, Manoj Plakal, Donald Beaver, Saul Jaspan, and Chandan Shanbhag. Dapper, a large-scale distributed systems tracing infrastructure. Technical report, Google, Inc., 2010. URL https://research.google/pubs/dapper-a-large-scale-distributed-systems-tracing-infrastructure/.

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

Yonggang Tao and Jingling Xue. Accelerating delta debugging through probabilistic monotonicity assessment. In *Proceedings of the 29th International Conference on Evaluation and Assessment in Software Engineering (EASE)*, pp. 12–22, 2025. doi: 10.1145/3756681.3756940.

Pablo Valle, Shaukat Ali, and Aitor Arrieta. Delta debugging for cyber-physical systems with flaky test executions. *arXiv preprint arXiv:2607.25695*, 2026. URL https://arxiv.org/abs/2607.25695.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer, 2005. doi: 10.1007/b106715.

Abraham Wald. Sequential tests of statistical hypotheses. *The Annals of Mathematical Statistics*, 16 (2):117–186, 1945. doi: 10.1214/aoms/1177731118.

Jason Walonoski, Mark Kramer, Joseph Nichols, Andre Quina, Chris Moesel, Dylan Hall, Carlton Duffett, Kudakwashe Dube, Thomas Gallagher, and Scott McLachlan. Synthea: An approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record. *Journal of the American Medical Informatics Association*, 25(3):230–238, 2018. doi: 10.1093/jamia/ocx079. Software (Apache-2.0), pinned commit 7e08387: https://github.com/synthetichealth/synthea.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

WARNER. Disease diagnosis dataset. Kaggle, CC0-1.0. https://www.kaggle.com/datasets/s3programmer/disease-diagnosis-dataset, 2024. 2,000 rows; accessed 2026-08-07.

Khushi Yadav. Synthetic medical symptoms & diagnosis dataset. Kaggle, MIT license. https://www.kaggle.com/datasets/khushikyad001/synthetic-medical-symptoms-and-diagnosis-dataset, 2025. 3,000 rows; accessed 2026-08-07.

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

Jiayi Yuan, Hao Li, Xinheng Ding, Wenya Xie, Yu-Jhe Li, Wentian Zhao, Kun Wan, Jing Shi, Xia Hu, and Zirui Liu. Understanding and mitigating numerical sources of nondeterminism in LLM inference. In *Advances in Neural Information Processing Systems*, volume 38, pp. 188045–188077, 2025. doi: 10.52202/085713-5653.

Andreas Zeller and Ralf Hildebrandt. Simplifying and isolating failure-inducing input. *IEEE Transactions on Software Engineering*, 28(2):183–200, 2002. doi: 10.1109/32.988498.

A Proofs

A.1 Proposition 1

The threshold scan seeks a score cutoff whose selected executions have population disagreement risk at most ε, with error probability at most δ. Fix the score rule and an ordered finite grid of candidate cutoffs $\lambda_1, \ldots, \lambda_L$ independently of calibration. Let the calibration tuples (S_i, E_i) be independent and identically distributed (iid), where S_i is the run-A score and $E_i \in \{0, 1\}$ indicates the declared disagreement event. For §3.3's loss, these tuples follow the usable-pair law conditional on U, the event that both answers are usable; availability and selection coverage retain their separate denominators. For each fixed cutoff with positive population selection probability, define $R_j = \Pr(E = 1 \mid S \leq \lambda_j)$, the selected count m_j, and the selected error count e_j. Conditional on $m_j = m > 0$, $e_j \sim \mathrm{Binomial}(m, R_j)$: selection truncates the common tuple law, and the selected labels retain its common marginal success probability. Conditioning instead on the observed score vector could give different success probabilities. The binomial law justifies using the one-sided Clopper–Pearson bound U_j, which satisfies $\Pr(U_j < R_j,\ m_j > 0) \leq \delta$.

Scan the cutoffs in the fixed order, stopping at the first $m_j = 0$ or $U_j > \varepsilon$, and return the last passing cutoff, or refuse if none passed. Let j^* index the first population-unsafe cutoff ($R_{j^*} > \varepsilon$), if one exists. Returning any unsafe cutoff requires the scan to pass the test at j^*, giving $U_{j^*} \leq \varepsilon < R_{j^*}$. This bound failure has probability at most δ. Thus the joint event of answering with selective population risk above ε has probability at most δ. No monotonicity of R_j is needed, and bounds at different cutoffs need not be independent. Skipping failed cutoffs and returning the largest passing one would invalidate this first-unsafe argument. ■

Scope. Empty empirical selection causes refusal; conditional risk at zero population coverage is undefined. The target averages risk over selected iid tuples, while a guarantee for an individual case and its already-issued answer would require a different analysis. Exchangeability alone is insufficient for the binomial step: $E_1 = \cdots = E_n = B$ with a single Bernoulli B is invariant to reordering but consists of one shared draw rather than n independent trials. Overlapping pairs share executions, and disjoint pairs may still share a case. Their pooled Clopper–Pearson bounds are nominal unless the requisite iid tuple model is justified (Appendix G).

Data-derived grids. A grid chosen from these same calibration scores does not acquire validity merely by conditioning on its realized value. An independently specified grid, an independent sample split, or a separate simultaneous-validity argument is required.

A.2 Proposition 2

Proposition 2 (non-monotonicity)**.** *Pair-disagreement risk (*PDR*) is not monotone in the set of predicates replaced by deterministic verifiers: the instance below exhibits a replacement that raises this risk from 0.17195 to 0.18. Thus a correct deterministic replacement need not improve agreement.*

Two independent executions of a decision that fires with probability p disagree with probability $2p(1-p)$, which is concave and maximal at $p = 1/2$. Consider a 1-of-n (OR) aggregator, which fires if any predicate returns true, over two independent predicates: A with firing rate $\theta_A = 0.9$ and truth 1, and B with $\theta_B = 0.05$ and truth 0. Mechanizing a predicate replaces it by its truth value, making it deterministic.

Unmechanized. The aggregate fires unless both predicates return 0, so $p = 1-(1-0.9)(1-0.05) = 0.905$ and $\mathrm{PDR} = 2(0.905)(0.095) = 0.17195$.

Mechanize B. B becomes the constant 0 and drops out of the OR, leaving $p = 0.9$ and $\mathrm{PDR} = 2(0.9)(0.1) = 0.18000$.

So $0.18000 > 0.17195$: replacing a stochastic predicate by a correct deterministic one increased disagreement. ■

The mechanism is counter-intuitive. B's occasional spurious firing pushes the aggregate from 0.90 to 0.905, further from $1/2$ and hence more stable. Mechanizing B removes that accidental stabilization; mechanizing A instead makes the OR constant and gives $\mathrm{PDR} = 0$. This counterexample does not make greedy strict improvement a necessary or sufficient allocation rule. The implemented rule rejects predicted non-improvements, so estimation error can reverse its realized effect, and a joint purchase may help even when no single purchase does.

A.3 SDD's Guarantee and Assumptions

Stochastic delta debugging (SDD) seeks a small set of perturbed factors that retains the observed divergence. Write $p(\Delta)$ for the divergence probability when perturbing subset Δ, and $\bar{\theta}$ for the pilot-derived threshold that defines sufficiency. For the pilot-dependent oracle $T^*(\Delta) = \mathbf{1}\{p(\Delta) \geq \bar{\theta}\}$, a nonempty set C is *1-minimal* if $T^*(C) = 1$ and $T^*(C \setminus \{x\}) = 0$ for every $x \in C$, including the empty deletion of a singleton. Thus the set is sufficient, and removing any single member makes it insufficient. *Inclusion-minimality* additionally requires $T^*(D) = 0$ for every proper subset $D \subset C$.

Strict mode. A sequential probability ratio test (SPRT) classifies a queried subset using repeated executions and likelihood-ratio stopping boundaries. Before sampling, declare a membership-test cap N_{cap}, error parameters $\alpha, \beta > 0$ with $\alpha + \beta < 1$, execution limits, and the conditional iid and separation assumptions below. Missing declarations cause refusal; their validity requires justi-

fication from the sampling design. Run the pilot and require the actual clipped endpoints to satisfy $0 < p_0 < \bar{\theta} < p_1 < 1$, otherwise refuse. Pruning and the subset-reduction algorithm `ddmin` propose a candidate. Every SPRT, including terminal checks, counts toward N_{cap}; refuse before test $N_{\text{cap}} + 1$, on any test timeout or sample-cap termination without a boundary decision, or on exhaustion of the execution budget. A nonempty terminal candidate must pass a fresh positive SPRT and every one-element deletion must pass a fresh negative SPRT. Only then apply the same empirical acceptance gate as in the historical procedure; return the candidate only if it passes. These terminal checks directly test sufficiency and single-member irreducibility, so the guarantee does not rely on pruning or search preserving them. In Figure 2, U is the original universe of perturbable factors, $n_{\max}$ limits executions per membership test, and B limits all executions, including pilot, pruning and gate draws.

StrictSDD$(U, N_{\text{cap}}, n_{\max}, B, \alpha, \beta)$

1. Require declared conditional sampling and separation assumptions. Set total membership count $N = 0$ and execution count $b = 0$.
2. Pilot on U to obtain $(\bar{\theta}, p_0, p_1)$. Refuse unless $0 < p_0 < \bar{\theta} < p_1 < 1$.
3. Define $\textsc{Test}(D)$: refuse if $N = N_{\text{cap}}$; increment N; draw at most $n_{\max}$ fresh executions until the SPRT crosses a boundary; otherwise refuse. Every execution, including pilot, pruning and gate draws, increments b and refuses before exceeding B.
4. Use optional read-closure pruning to propose $V \subseteq U$. Set $C = \texttt{ddmin}(\textsc{Test}, V)$.
5. Refuse if $C = \varnothing$ or a fresh $\textsc{Test}(C)$ is negative. For every $x \in C$, refuse if a fresh $\textsc{Test}(C \setminus \{x\})$ is positive.
6. Apply the empirical acceptance gate using controls from the original universe U. Return C only if the gate accepts; otherwise refuse.

SelectiveScan$(\{\lambda_j\}_{j=1}^{L}, \varepsilon, \delta)$

1. Fix the score rule and ordered grid independently of calibration. Initialize the output to refusal.
2. For each λ_j in order, compute selected count m_j and a valid upper bound U_j. Stop if $m_j = 0$ or $U_j > \varepsilon$; otherwise set the output to λ_j.
3. Return the output with its sampling contract and coverage. Run separately per stratum for per-stratum confidence.

Figure 2: Procedural specification. Refusal aborts the current procedure. The SPRT uses likelihood-ratio boundaries $(1 - \beta)/\alpha$ and $\beta/(1 - \alpha)$. Enforced stopping rules do not verify the declared statistical assumptions.

Proposition 3 (SDD, 1-minimality under a stochastic oracle)**.** *Conditional on the pilot and the history before each adaptively chosen query, assume fresh iid Bernoulli executions with fixed query probability $p(\Delta)$, and $p(\Delta) \le p_0$ or $p(\Delta) \ge p_1$ for every set reachable under the adaptive procedure, including terminal checks. If these assumptions hold almost surely over pilots and histories on which strict mode proceeds, then*

$$\Pr\{\textit{strict SDD returns a set not 1-minimal for } T^*\} \le \delta := N_{\text{cap}} \max\left\{\frac{\alpha}{1-\beta}, \frac{\beta}{1-\alpha}\right\}.$$

No monotonicity assumption is required. The bound is joint, not conditional on returning a set.

Proof. Given a query's pre-test history, let L_n be the likelihood ratio of iid Bernoulli(p_1) to Bernoulli(p_0) observations. If $p \le p_0$, L_n is a nonnegative supermartingale starting at 1, so the probability of crossing $(1 - \beta)/\alpha$ is at most $\alpha/(1 - \beta)$. If $p \ge p_1$, L_n^{-1} is such a supermartingale, bounding a crossing of $\beta/(1 - \alpha)$ by $\beta/(1 - \alpha)$. Aborting on limits only removes boundary decisions. By conditional expectation and a union bound over the N_{cap} possible test slots (unused slots have no error), the probability of any wrong boundary decision is at most δ. If no decision is wrong, the terminal checks establish both sufficiency and every required negative deletion. The acceptance gate can only remove that candidate, so cannot enlarge the joint bad-return event. Averaging over the pilot gives the stated bound. ■

What is enforced, and what remains assumed. Strict mode enforces the cap, endpoint order, boundary-only decisions, and terminal checks. None establishes separation: a probability inside

(p_0, p_1) can still reach an SPRT boundary. Nor does the pilot establish conditional iid sampling or separation from its random clipped endpoints. If these conditions are justified only for particular pilots, the guarantee is restricted to those pilots; an unconditional claim needs them almost surely or a separate bound on their failure. D5 deliberately challenges separation and is not made theorem-valid by lower α or cap refusal. The theorem establishes minimality for T^*; interpreting that set as a physical cause additionally requires the oracle to represent the causal target. The acceptance gate's two-proportion comparisons at $\alpha_{\text{cert}} = 0.05$ filter candidates empirically. They do not test the sampling or separation assumptions.

Joint versus conditional risk. If the assumptions hold and the return probability is $c > 0$, the bound implies conditional non-minimality at most $\min(1, \delta/c)$, not δ. It bounds neither refusal frequency nor realized error frequency in a finite corpus. Realized $N\alpha$, $\bar{N}\alpha$, and observed maxima are workload diagnostics, never substitutes for the predeclared N_{cap} in this guarantee.

1-minimality, measured separately. Table 6 audits the historical, non-strict procedure on 1,500 candidates, including candidates later refused by the gate. It estimates T^* from 300 fresh executions per subset on a random-number-generator (RNG) stream independent of the search. Sufficiency and irreducibility are separate measurements; the strict theorem bounds failures of either. The audit marks a candidate boundary-limited if any estimated rate is within 0.05 of $\bar{\theta}$, excluding those from its clear-violation count. This finite Monte Carlo audit is not a proof of population 1-minimality.

Monotonicity and separation are distinct. The concrete counterexample

$$p(\varnothing) = 0.02, \quad p(\{a\}) = 0.8, \quad p(\{a, b\}) = 0.02, \quad p(\{a, b, c\}) = 0.9$$

has population pilot threshold $\bar{\theta} = 0.45$ and, at $\gamma = 0.45$, unclipped endpoints $p_0 = 0.2475$, $p_1 = 0.6525$ (the clipping leaves them unchanged). All displayed probabilities are separated, yet $\{a\} \subset \{a, b\}$ crosses from positive to negative. Set the other nonempty proper subsets' probabilities to 0.02: then $\{a, b, c\}$ is 1-minimal but not inclusion-minimal because $\{a\}$ is sufficient. Thus the pilot does not make non-monotonicity equivalent to separation failure. If T^* is monotone, 1-minimality does imply inclusion-minimality: any proper sufficient subset would lie inside some one-element deletion, making that deletion sufficient too. Even then minimum cardinality among all sufficient sets is a stronger, unproved claim.

Table 6: Historical candidate 1-minimality audit, both calibrations, on 1,500 instances (300 per tier). The last column counts candidates failing sufficiency or irreducibility outside the stated boundary margin. These are not strict-mode results.

α	D1	D2	D3	D4	D5	overall	clear violations
0.02, as shipped	1.000	1.000	0.990	0.867	0.120	1,193/1,500	124
$0.1/334$, historical	1.000	1.000	1.000	0.933	0.170	1,231/1,500	14

The tighter historical calibration reduces clear audit violations from 124 to 14; it does not validate the theorem's assumptions. At $\alpha = 0.02$, the estimated 1-minimality rate is 0.981 among 1,203 gate-accepted candidates and 0.044 among 297 withheld candidates. The gate therefore selects candidates with substantially different audited properties on this corpus, but still accepts imperfect candidates (D5's rate is 0.583) and rejects some 1-minimal ones (four on D2). With one uniform cap $N_{\text{cap}} = 334$, the shipped error parameters give $N_{\text{cap}}\alpha/(1-\alpha) > 1$ on every tier, regardless of its realized test count. The bound is therefore vacuous: it provides no improvement over the trivial probability bound of one.

B Reproducibility Details

Result provenance. Analysis scripts (*producers*) compute the reported numbers from recorded evidence or declared simulations. Recorded evidence is authoritative for historical values; corrected producers can change regenerated results. The matched-attribution, paired-uncertainty and cluster-bound analyses are committed under `experiments/results/`. The anonymous supplement includes these analyses, the distributed recorded evidence and the evidence-consistency suite; its manifest identifies the included files and checksums. The regeneration requirements below identify artifacts that are not distributed.

Artifact map. Scripts in `experiments/` write analyses to `experiments/results/`. Each experiment below names its producer; examples include `campaign_b_identical_inputs` and `campaign_b_window_lead` for input variation, `sdd_calibration` for membership-test counts, and `toolloop_overhead` for recording cost. `experiments/README.md` lists regeneration limits.

The standing suite writes `seam/results/results.json` and `evals/results/`. Five evaluation-seed recordings back the tamper-localization counts in Appendix G. Recorded evidence permits statistical reanalysis; unseeded races, machine timings and corrected implementations can change regenerated experiments.

CAMELYON regeneration. Code, configuration and executed notebooks are committed. `11_seam_determinism_before_after` contains the arms, cascade and margins; `09_innovation_ablations` contains the full-slide associativity probe. Both use detector `472c3e06...`. The supplement includes the original arm/sealing summaries, `determinism_arms_a3.json` and `lesion_truth_by_arm.json`. Tile caches, detector weights and full run recordings still require regeneration: public archive range reads (about 300 MB cold, no whole-slide download) and retraining on the 460 recorded patches. Reproducing the weight digest additionally requires the verified environment below; a laptop execution alone does not establish that digest.

Loop reanalysis and replay. The four default recordings contain 2,056 billed model calls. Offline model-response reuse has 0 model misses and 0 provider calls in all four; the original tool replay check fails on one draw-4 case. Re-driving the unsealed arm instead produces 71 misses in 72 episodes because its schedule was not recorded. These distinct replay limits are detailed below. The local fixed-selector control and historical attribution corpus regenerate without provider calls, with recorded runtimes of about half a minute and a minute, respectively.

B.1 Environment, Seeds and Data

CAMELYON environment. Python 3.13.5, `torch` 2.13.0+cu126 (CUDA 12.6, cuDNN 91002) on 4× NVIDIA L40S, `numpy` 2.1.3, `scipy` 1.16.1 and `tifffile` 2026.7.31. The reconstruction matches all 11 environment fields fingerprinted by the image harness and retrains detector weights with bitwise agreement. The repository's `pyproject.toml` pins the general experiment dependencies; these additional image-stack versions specify the detector regeneration environment.

Seeds and configuration. One row per campaign; the [R] campaigns additionally pin a model id and a committed recording, named with the campaign below.

Table 7: Seeds and configuration. One row per campaign; the [R] campaigns additionally pin a model id and a committed recording, named with the campaign.

Component	Seed / setting
§4.4 attribution suite	master seeds 0–4 (five committed runs), profile `full`, 300 instances per tier; seed 4 supplies the historical tighter-calibration column in Appendix C
§4.1 CAMELYON	seed 42, 6 repeats, pyramid level 4 (a reduced-resolution slide image), profile `smoke`
§4.1 detector weights, all of Table 2	SHA-256 `472c3e06da408a3d11963e0a3b59dcb3e119a9ff71191b19e8dbcf38c91dfaca` (bitwise agreement on two retrains)
§4.4 SPRT (SDD's membership test)	$\alpha = 0.02$ as shipped; Appendix C distinguishes the tighter historical and corrected strict calibrations

Data access. CAMELYON17 is read anonymously from the Registry of Open Data on AWS (AWS Open Data Registry, n.d.) (unsigned requests) over HTTP range requests; no credentials are read or stored and no whole slide is fetched. Each slide's level-4 plane costs 13–18 MB, so the 20-slide study transfers about 300 MB cold and nothing on re-runs, against a 1,000-slide, 2.7 TiB archive. The `tifffile` parse of a 2.8 GB pyramidal BigTIFF header uses 8 range requests and about 1.17 MB; the full slide open, which walks every image file directory (IFD), uses 66 requests and about 1.57 MB. The 13–18 MB per slide is the level-4 tile cache, built once.

Campaign A data (§4.3). A fixed slice of the `mteb/banking77` test split at revision `18072d26` (mteb, 2025).

Campaign B case provenance (§4.3). Cases are selected without replacement by cycling through diagnosis labels after seeded within-label shuffles, giving near-equal diagnosis representation rather than preserving corpus prevalence. They come from the supplemental corpus in `evals/datasets/medical_diagnosis/kaggle_instances.jsonl`, normalized by committed code from two synthetic Kaggle case datasets under CC0-1.0 and MIT (WARNER, 2024; Yadav, 2025). A third MIT-licensed disease-to-treatment dataset supplies exact diagnosis-matched treatment descriptions (Sanaei, 2025); its synthetic provenance is unspecified. All three were accessed 2026-08-07, with archive SHA-256 values pinned in the builder. The directory's separate canonical corpus is generated from a pinned Apache-2.0 Synthea commit (Walonoski et al., 2018). Provenance, licenses and the rejected candidates — including four Hugging Face datasets excluded for schema or extraction reasons — are recorded row-by-row in its `sources.jsonl`.

Population and sampling scope. Campaign A is a fixed benchmark slice, not an iid sample of future banking traffic. Campaign B samples without replacement from the specified finite synthetic corpus; a seed fixes case selection, not a superpopulation model or independence of provider responses. The reported case-level CP calculations posit iid case events from a hypothetical workload population with the evaluated case mix, under a fixed model, prompt policy and repeat protocol. They are not exact design-based bounds for the finite sampling scheme. Hoeffding likewise needs independent bounded case observations. We therefore use these bounds to audit what the stipulated sampling models would support, not to certify deployment on new populations or provider epochs. When availability varies by case, pooling only usable pairs also changes the case weights; an equal-case analysis and a usable-pair analysis need not share an estimand.

The tile budget (§4.1). The 6,000-tile budget models a deployment cost cap and binds on this cohort, allowing admission order to affect which tiles are processed. At pyramid level 6 only 598 candidates survived the tissue filter and a 4,000 budget dropped nothing; that setting would not exercise budget-driven truncation. The runner reports per slide whether the budget truncates the candidate set.

Table 2's denominators (§4.1). The `/100` in rows 1–2 is 20 slides $\times$ 5 comparisons against repeat 0. The six-run pN null uses 15 consecutive-run comparisons from three stageable challenge groups; five further pairs concern the refused group. A binomial bound treating the 15 comparisons as independent is nominal. Under an iid group model, zero groups with any stage flip out of three gives a one-sided 95% case-any bound $1 - 0.05^{1/3} = 0.632$, a different target. The separate M4 lesion-class cascade resets seeds to 42 and 1042 for each slide and compares those two executions; millimeter-based measurements use the 15 slides with a micron scale.

Single-detector comparison (§4.1). Every row of Table 2 uses `472c3e06...`. Two retrainings under the declared environment, data, code and seed achieve bitwise agreement in the detector weights. These dependencies jointly define the reproducibility conditions. The unsealed lesion-class flip count is 29 and the threshold-margin populations are 1,435/1,071 (Appendix D.1).

Cache identity and environment resolution. Detector probabilities in `artifacts/tile_cache/` were keyed on `{slide}_L{level}_p{patch}_s{stride}` without a weights digest, permitting cached scores from a different detector to be reused silently. The sibling `mil_bags_L4.npz` checked its digest and refused the mismatch. The tile cache now stores and verifies a digest, fails closed on a mismatch, and reports caches written before the fix as unverified (§5). Separately, the Jupyter kernelspec resolved a bare `python` from `PATH`; this could select a different interpreter and is now pinned to an absolute path. These cache-identity and environment-resolution failures motivate recording dependencies beyond the visible model request.

The associativity probes (Appendix D.2). The two detector digests gave the same reported maximum deltas (1.19×10^{-7} and 5.96×10^{-8}); the mean absolute delta changed. This is an observation on these probes, not score-independent invariance: floating-point spacing and accumulation error depend on magnitudes and cancellation. A1's measured size-class flip count is 0.

The §4.2 tool-loop workload. The loop stops when satisfied or at a budget cap of 8 calls with at most 2 retries, over a 6-shard ledger with a 3-candidate window. A deterministic final adjudication rule isolates outcome divergence arising in the preceding execution. Shard latency is drawn from an unseeded `random.Random`, producing the unsealed arm's arrival-order race. L_{tool} counts tool attempts, including retries, and is bounded by $8 \times 3 = 24$: each budgeted call permits an initial attempt and its retries. Two kinds of campaign are reported: four independent billed draws [R] (510,

519, 513 and 514 model calls on `anthropic.claude-sonnet-5`, with the model choosing each step at a 256 output-token cap; all four recordings committed) and a fixed-selector control [M] (always choosing the first candidate, no provider call, 30 draws; Appendix D.5).

Recording cost. `experiments/toolloop_overhead.py` measures ten fixture draws with 720 episodes per arm and three arms interleaved per (case, repeat) [M]. Episode timing uses `perf_counter`; directory sizes use `os.stat`. Neither enters the trace. Median ordered-only execution takes 0.150 ms per episode or 0.038 ms per tool step. Recording takes 2.64 ms per episode or 0.66 ms per step, including 0.61 ms inside the gateway. Verification and replay add 0.323 and 0.113 ms per step. Recording is thus 17.5× ordered-only compute, and the full cycle 31×, but recording is 0.03% of the observed live-call latency (about 2.2 s; 515 calls in 1,150 s).

Storage and comparison scope. Each step writes 137 bytes in two content-addressed objects, 485 in the trace and 158 in the index: 780 logical bytes. Allocation is 10 KiB on this filesystem because each roughly 70-byte object occupies a 4 KiB block. These timings and allocated sizes are machine-dependent. The unsealed arm spends 98% of its time in simulated shard delays of up to 1.8 ms, so its latency ratio is not a recording-overhead estimate.

Decode configuration. Billed campaigns send no sampling parameters. All three model IDs reject `temperature` with HTTP 400, recorded on 2026-08-05 in each artifact's `temperature_note`; the API documents no seed. On 2026-09-05, `experiments/decode_probe.py` also records rejection of `top_k=1` and `top_p=0.01` by the loop's model. These controls cannot supply a provider-side greedy arm. The same probe finds that `thinking: disabled` is accepted by the loop's model and rejected by Campaign A's; it does not establish support across all models.

Default campaigns leave reasoning enabled. Output caps are 256 tokens in Campaign A's epochs 1, 2 and 4, 1,024 in epoch 3 and Campaign B, and 256 per loop model call. Campaign A's epochs 3–4 additionally store per-response `stop_reason` and token counts. Manifests record `sampler_determinism: unsealable` and `temperature: null`; these fields record configuration limits, not a measured decomposition of provider variation. Appendix D.7 separately reports the reasoning-disabled loop draw.

Model compatibility (§4.2). To select a model that returns usable tool choices on this workload, we probed all three models over four prompt variants: step 0 and mid-episode, each with real `INV-...` candidate ids or generic stand-ins. The probe uses 4 repeats per variant, totaling 48 billed calls.

Table 8: Model compatibility on four variants of the tool-loop prompt, 4 repeats each, 48 billed calls. Empty answers contain no text usable as a tool choice.

model	empty answers	reading
`claude-sonnet-5`	0 / 16	usable on every variant
`claude-fable-5`	12 / 16	usable on one variant only
`claude-opus-5`	16 / 16	empty on every variant

The probe's empty responses use only 1–10 output tokens, below the 256-token cap; usable responses use 7–250. The empty responses recur on a second probe, establishing a compatibility problem below the cap while leaving the internal provider mechanism unidentified. Campaign B's 1,440 opus calls have 0 failures under a different free-text diagnostic prompt, demonstrating the prompt-specific scope of the loop result. A campaign attempt on the cheapest model produced 8 unusable responses in 17 model calls and no completed episode. The reported sonnet campaign has 4 unusable responses in 2,056 calls.

Output-cap evidence (§4.2). Draws 3 and 4 record token counts over 1,027 model calls. Draw 3 has median 10, 95th percentile 99 and maximum 256 tokens, with one call reaching the cap and one returning an empty answer. The recorded aggregates do not establish whether these were the same call. Draw 4 has 0 unusable answers and 0 at-cap calls in 514 model calls, with median 44, 95th percentile 95 and maximum 215 tokens. The combined at-cap count is 1 in 1,027. The cause of draw 3's empty answer remains unresolved, including whether cap exhaustion contributed; Table 8's low-token empty responses come from a separate probe. Current instrumentation stores token counts within unusable-response records to make this association directly checkable. Recording these details supports diagnosis when failures occur; it does not prevent the failures.

B.2 Recording Provenance and Replay Contract

The following notes identify what the recorded evidence supports, how failures are handled, and which checks require a trusted recording anchor.

Transient provider failures (§4.2). An interrupted full campaign stopped at call 237 of a possible 1,152 on a `HTTP 529 overloaded_error`, after the preceding answers had been billed. The harness preserves a partial recording for that case. The model boundary retries transient statuses (429/5xx) with exponential backoff and fails fast on request or authorization errors (400/401/403). This policy limits recovery attempts to statuses treated as transient. The reported campaign completed in 1,185 s with no unretried failure.

Historical draw-4 replay failure (§4.2). In draw 4 the sealed case `INV-0004` reports `replay_identical: false` at 56 gateway calls against 60 recorded tool executions. `all_traces_verified` stays true and `replay_tool_executions_after_record` stays 0, so internal trace consistency and absence of live tool calls hold while the tool-sequence replay check fails. The replay check compares tool sequences and also returns false on a caught replay exception; full-trajectory bitwise agreement is a separate property. The failure cause cannot be recovered because the reason was not persisted.

To test whether the failure recurs from the recorded evidence, `experiments/draw4_replay_probe.py` re-drives the campaign from draw 4's committed cache over 30 trials, without provider calls.

- All nine compared sealed-arm aggregates reproduce the committed artifact in 30 of 30 trials. The per-case statistics for `INV-0004` also reproduce: 4 distinct tool sequences over 3 distinct lengths, range 4–6, 12 retries, 60 tool executions. This case has the greatest tool-sequence diversity in the corpus; no other case in any recording exceeds 2 lengths or 3 tool sequences.
- Replay succeeds in 30 of 30 trials, serving all 60 gateway calls.
- No `(tool, args)` key occurs more than once in any of the twelve episodes. A last-wins index and an occurrence-indexed index therefore return the same bytes on this recording, so the current gateway's occurrence lookup does not alter this re-execution comparison.

The recorded result remains one replay failure in 288 sealed episodes, unreproduced in 30 controlled re-executions. The absence of repeated keys excludes a repeated-key collision as its explanation; the available evidence cannot identify an exception or another fault outside the recording. Table 3 therefore retains successful tool replay in 3 of the 4 default-decode draws, or 4 of 5 including the temporal replication. Current instrumentation detects and records replay failure reasons for diagnosis. It neither recovers the missing historical reason nor prevents every replay failure.

Current replay contract (§4.2). The tool gateway indexes each canonical `(tool, args)` request hash to an ordered list of response hashes; a per-key cursor selects the next recorded occurrence, and a fresh gateway starts each replay pass. Responses live in the content-addressed store (CAS) under their own content hashes. Missing requests and exhausted occurrences produce distinct refusals. The model cache likewise selects recorded occurrences, and the experiment harness walks the recorded trace. The loop's attempt-bearing arguments remain part of its historical recordings, although occurrence lookup no longer depends on adding such arguments. Offline replay supplies only recorded occurrences and makes no fresh model draws.

Four reproducibility activities. Statistical reanalysis uses recorded evidence to preserve the finite observations, including the unsealed loop column. Offline replay reuses recorded interactions without provider calls: the four default loop draws reproduce sealed aggregates with no model misses, while the original draw-4 tool replay reported one unexplained failure. Re-driving the unsealed loop with its unrecorded race produces 71 misses in 72 episodes in the reported check; replaying that race would require recording and enforcing its schedule. Regeneration reruns seeded local experiments under their declared environments. Replication collects fresh billed calls to test whether observations recur. Fresh-run stability is evaluated by replication, while the other activities check computations or recorded executions.

Completeness anchors. A writer can rewrite and rehash an internally consistent trace. Detecting such changes requires a trusted digest or signed checkpoint whose signing authority is protected from that writer. In `truncation_anchor.py`, tail truncation and terminal re-chaining escape the bare trace checker in 12 of 12 cases and are caught by a signed length-and-terminal-digest checkpoint in 12 of 12. The sweep uses run lengths 1, 2, 3, 4, 8 and 16, with interior re-chaining as a control. An anchor authenticates the recording at signing time; it does not establish that the recorder was honest.

Checkpoint coverage. The runner and experiment harness write `checkpoint.json` when a signer is configured, binding `event_count` and `terminal_hash`; otherwise they write `checkpoint.absent.json`. The LangGraph adapter writes automatically on exit from `SealedApp.invoke`; direct tool-node calls such as the loop's bypass that checkpoint-writing path. Bare `SealedGraph.verify()` returns a boolean internal-consistency result and can accept a truncated prefix. It has no explicit `UNCHECKED` completeness state, so callers must inspect checkpoint availability separately.

CAMELYON check. The regenerated sealing artifact is anchored; the separately re-declared attribution runs in Appendix C.2 are not. Truncating the anchored trace yields `verify_ok: true` but `anchored_verify_ok: false`, because the surviving 19 events do not match the signed count of 20. The artifact records `anchored: true` and `detected_by_checkpoint: true`. Reproducing that original signature requires its private key, which is not distributed. A new signer can test the same mechanism, as the local 12-of-12 sweep does, but cannot authenticate the original recording.

GPU-kernel control. Table 2's arms reuse cached detector outputs, holding kernel effects fixed. A separate probe runs the kernels on 4× L40S: 8/8 forward passes have bitwise agreement with `max_abs_delta` exactly 0.0, under both the default and `use_deterministic_algorithms(True)`. The probe covers fixed-batch, in-process forward inference of the 101k-parameter convolutional neural network (CNN). Varying batch shapes, backward reductions and cross-process `cudnn_benchmark` fall outside that test.

Four audit failure modes (§5). (i) A model-boundary index that retains only the last response for a request loses occurrence identity, yielding bitwise agreement on 0.891–0.921 of calls. Occurrence-aware lookup reaches 1.000 and refuses over-consumption distinctly (Appendix E). (ii) Failure records without reasons prevent diagnosis. Current instrumentation persists reasons, but the four default recordings predate that instrumentation, leaving draw 4 unattributed. (iii) A tile cache without the detector-weight digest cannot verify detector identity. Current reads verify the digest and fail closed on a mismatch; pre-fix caches remain unverified. (iv) An unanchored chain permits undetected tail removal. Completeness checking requires a configured signer and trusted checkpoint; unanchored recordings retain that limitation. The model-boundary defect in (i) does not explain draw 4's tool-boundary failure, where no key repeats within an episode.

C SDD Calibration, Baselines and the Real Divergence

This section examines how membership-test calibration affects attribution recovery, refusals and execution cost. Proposition 3 applies to the optional strict mode, with a predeclared cap N_{cap} counting search and terminal SPRTs. The tables below retain non-strict runs and post-hoc refusal filters, whose coverage and cost must be distinguished from strict-mode measurements. Table 5 and its strict supplement come from `experiments/results/ddmin_matched.json`, produced by `experiments/ddmin_baseline.py --matched`. Realized $N\alpha$ and $\bar{N}\alpha$ describe test workload; the theorem uses the predeclared cap to bound error.

Counters. A membership test resolves one subset query by SPRT. Historical N counts only `ddmin` search tests; strict mode's N_{cap} additionally covers the fresh terminal candidate and deletion tests. A workload execution is one run under a perturbation, including pilot, pruning, membership, and gate draws when reporting end-to-end cost. An instance is one attribution problem, 300 per tier per seed. The synthetic attribution corpus is [M] and makes no model calls; executions here are not the tool-loop episodes of §4.2.

Historical constants and gate. The pilot uses $n_{\mathrm{pilot}} = 40$ executions at $\Delta = \mathrm{ALL}$ and sets $\bar{\theta} = \frac{1}{2}\max(\hat{p}(\mathrm{ALL}), 10^{-3})$. With $\gamma = 0.45$, $p_0 = \bar{\theta}(1-\gamma)$ is clipped to $[0.02, 0.9]$ and $p_1 = \bar{\theta}(1+\gamma)$

Table 9: Historical SDD, five seeds with ranges, 300 instances per tier per seed. Exact match is against the injected minimal-cause family. The last column applies a post-hoc capped-test filter to the historical tighter-calibration output (seed 4); it omits strict-mode terminal checks and does not establish the theorem's assumptions.

tier	coverage	conditional exact match	historical cap-filtered coverage
D1	1.000	1.000	1.000
D2	0.996 [.987, 1.000]	1.000	0.987
D3	0.997 [.993, 1.000]	0.999 [.997, 1.000]	0.997
D4	0.884 [.867, .910]	0.998 [.996, 1.000]	0.733
D5	0.181 [.160, .200]	0.963 [.940, .983]	0.017

to $[p_0 + 0.02, 0.95]$. The historical settings are $\alpha = \beta = 0.02$ and $\alpha = \beta = 0.1/334$, both with $n_{\max} = 150$. A test without a boundary decision at that limit returns $\hat{p} \geq \bar{\theta}$, which has no asserted SPRT error guarantee. Pruning uses eight further executions at ALL and retains factors readable at or before the latest observed first-divergence step. The acceptance gate uses $n_{\text{cert}} = 60$ executions per evaluated set, comparing the candidate with the median of three size-matched random controls and with every one-element deletion through one-sided two-proportion tests at $\alpha_{\text{cert}} = 0.05$. This empirical filter does not certify separation, cause correctness, or a familywise error rate.

Timeout filtering is not strict-mode validation. At the tighter historical setting, 567 of 30,917 D4 tests and 3,899 of 34,504 D5 tests end at the sample cap. This affects 62 of 282 D4 answers and 57 of 62 D5 answers. Filtering out any answer with a capped test leaves historical coverage 0.733 and 0.017 respectively (Table 13). It does not add the endpoint-order check or fresh terminal rechecks, and it cannot establish conditional iid sampling or separation. In particular these D5 records are not covered by Proposition 3 merely because the remaining tests reached boundaries.

Table 10: Five difficulty tiers of the synthetic attribution oracle (§4.4). k factors are read over T steps; $|C|$ is the injected factor-set size and the trigger specifies which members must be perturbed to activate the added divergence hazard q_{hi}. On D2 either member alone is a minimal cause. q_{lo} is the total background divergence probability with every factor perturbed. From `seam/drift_bench.py` and `seam/bench/MANIFEST.json`.

tier	k	T	trigger	$\lvert C\rvert$	q_{hi}	q_{lo}
D1	16	12	single	1	0.90	0.02
D2	32	16	disjunctive (any member arms)	2	0.75	0.06
D3	64	20	conjunctive (all members arm)	2	0.55	0.15
D4	64	24	conjunctive	3	0.35	0.18
D5	64	24	conjunctive	3	0.22	0.30

Historical counter and cap filter. Tables 11 and 12 record uncapped `ddmin_probes`, then rewrite instances with $N \geq N_{\text{cap}} = 334$ as refusals. Thus a recorded maximum of 410 is compatible with every retained answer having $N \leq 333$. The output filter matches a corresponding historical search abort, but its cost includes work after the abort point. It is not the strict procedure, whose total test cap also counts terminal rechecks and whose per-test limits abort immediately. No new strict-mode costs or coverage are inferred from these filtered records.

Execution cost. Historical SDD uses a mean of 378 / 394 / 1,101 / 2,501 / 3,403 workload executions per D1–D5 instance, versus 424 / 808 / 1,576 / 1,576 / 1,576 for one-factor-at-a-time testing. SDD costs less on D3 and more on D4–D5, while the one-factor baseline returns no answers on those tiers. These counts measure local workload executions; a billed application would need to budget the model calls required by every such execution.

Exact calibration constants. For strict mode with $N_{\text{cap}} = 334$, setting

$$\alpha = \beta = \frac{0.1}{334 + 0.1} \quad \text{gives} \quad \frac{334\alpha}{1 - \alpha} = 0.1$$

Table 11: Historical uncapped search-test counts over 300 instances per tier at $\alpha = 0.02$. Products $\bar{N}\alpha$ and ratios $0.10/\max N$ are workload and calibration diagnostics, not error bounds. The observed maximum informed a cap for subsequent runs; it is not a predeclared per-tier cap for this run.

tier	mean N	median	p90	max N	$\bar{N}\alpha$ at $\alpha = 0.02$	$0.10/\max N$
D1	5.2	5.0	6	8	0.10	1.3×10^{-2}
D2	6.2	6.0	7	9	0.12	1.1×10^{-2}
D3	60.8	65.0	80	122	1.22	8.2×10^{-4}
D4	105.9	104.0	134.1	311	2.12	3.2×10^{-4}
D5	111.4	97.0	187.1	334	2.23	3.0×10^{-4}

exactly, subject to Proposition 3's assumptions. The historical artifact instead uses $\alpha = \beta = 0.1/334$, whose same cap expression is 0.10002995. Literal $\alpha = \beta = 0.000299$ would give 0.09989587; it is not the stored constant. Rounding these settings to the same printed value conceals the distinction. None of these numerical calculations establishes separation.

Table 12 reports the full 1,500-instance corpus under both calibrations, on the same instances and seeds, so the contrast is not confounded by a change of population.

Table 12: Historical calibration comparison on master seed 4. The tighter setting is exactly $\alpha = \beta = 0.1/334$. N counts its uncapped search tests, executions include the pilot, pruning and gate, and answer rates apply the post-hoc $N \geq 334$ refusal filter. $\bar{N}\alpha$ is diagnostic. These are not strict-mode measurements.

tier	answer rate $\alpha = 0.02$	answer rate $\alpha = 0.1/334$	mean N	max N	$\bar{N}\alpha$	executions	cost ratio
D1	1.000	1.000	5.2	8	0.0016	378 → 406	1.07×
D2	0.987	0.987	6.2	9	0.0019	394 → 437	1.11×
D3	0.993	1.000	60.7	83	0.0182	1,101 → 1,812	1.65×
D4	0.870	0.940	103.1	305	0.0309	2,501 → 4,448	1.78×
D5	0.160	0.207	115.0	410	0.0344	3,403 → 7,264	2.13×

Table 12 separates changes in test workload, answer rate and cap-filter behavior.

Test-count products decrease. The largest tier mean product $\bar{N}\alpha$ falls to 0.0344. This describes the test workload at the tighter calibration, not a realized error probability, and does not show that Proposition 3's assumptions hold.

Higher coverage at higher cost. Historical answer rates rise from $0.993 \to 1.000$, $0.870 \to 0.940$ and $0.160 \to 0.207$ on D3–D5. The tighter setting also increases tests reaching the 150-execution cap and using the threshold-vote fallback (Table 13). The observed coverage gain therefore cannot be interpreted as stronger guaranteed recovery. The nominal boundary scale changes by $\log(1/3.0 \times 10^{-4})/\log(1/0.02) \approx 2.1$, close to D5's measured $2.13\times$ cost ratio; this comparison does not derive the cost of the adaptive search.

The historical cap filter fires. At $\alpha = 0.1/334$, two of 300 D5 instances have $N \geq 334$, with maximum 410. Both already fail the empirical acceptance gate, so the recorded answer rate remains 0.2067 before and after this filter. This observation says nothing about the extra refusals or cost of the strict procedure's terminal checks.

Attribution metrics. We abbreviate cause-family exact match as MCA (minimal-cause agreement) in artifact-facing tables. Let $\mathcal{M}_i$ be instance i's planted minimal-cause family, C_i its search candidate, $\widehat{C}_i$ its returned set (empty on refusal), and $A_i = \mathbf{1}\{\widehat{C}_i \neq \varnothing\}$. D2 with disjunctive planted factors $\{a, b\}$ has $\mathcal{M}_i = \{\{a\}, \{b\}\}$, not $\{\{a, b\}\}$. Conjunctive tiers have the whole planted cause as their sole family member. Define

$$\mathrm{MCA}_i(D) = \mathbf{1}\{D \in \mathcal{M}_i\}, \qquad F_i(D) = \max_{G \in \mathcal{M}_i} \frac{2|D \cap G|}{|D| + |G|},$$

with both scores zero for refusal. Candidate MCA is $n^{-1}\sum_i \mathrm{MCA}_i(C_i)$ before the gate. Coverage is $c = n^{-1}\sum_i A_i$; conditional exact recovery is $\sum_i A_i \mathrm{MCA}_i(\widehat{C}_i)/\sum_i A_i$, whereas unconditional exact recovery is $n^{-1}\sum_i \mathrm{MCA}_i(\widehat{C}_i)$. Conditional and unconditional max-F1 replace MCA by F_i in these expressions. Conditional scores are undefined at zero coverage. Here "cause agreement" means conditional MCA; it is distinct from threshold-oracle 1-minimality.

The benchmark's separate *prescribed-action agreement* is $n^{-1}\sum_i \mathbf{1}\{A_i = b_i\}$, with $b_i = 1$ on D1–D4 and $b_i = 0$ on D5. It measures compliance with that declared answer/refuse policy, not cause accuracy or method-independent identifiability. A D5 answer can violate the policy and still recover a planted cause. The one-factor baseline's 900 policy credits per 1,500 instances are 300 D1 answers, 300 D2 answers and 300 D5 refusals; its five-seed D2 conditional MCA is only 0.6%. Its observed zero coverage on D3–D5 does not establish impossibility for other methods.

Table 13: Historical tests terminating at $n_{\max} = 150$, seed 4, from `experiments/sdd_calibration.py`. Answers already apply the $N \geq 334$ filter; the last column additionally removes any answer with a capped test. This post-hoc diagnostic is not strict-mode coverage and does not establish separation.

α	tier	capped / tests	answers	answers with a capped test	answer rate, refusing those
0.02	D1–D3	0 / 21,686	894	0	unchanged
0.02	D4	69 / 31,776	261	11	0.833
0.02	D5	285 / 33,433	48	12	0.120
0.1/334	D1–D2	0 / 3,435	596	0	unchanged
0.1/334	D3	1 / 18,212	300	1	0.997
0.1/334	D4	567 / 30,917	282	62	0.733
0.1/334	D5	3,899 / 34,504	62	57	0.017

C.1 Contrast membership testing under high background divergence

D5's high background divergence can obscure an injected cause under an absolute probability threshold. To test whether comparison with a control set improves recovery, `experiments/contrast_membership.py` replaces only the membership predicate, retaining the shipped pilot, pruning, `ddmin` and gate ([M], seed 4, 300 instances per tier). Each execution of Δ is paired with an execution of a fresh size-matched control set. A *discordant pair* is one in which exactly one execution diverges. Conditional on discordance, the probability π that Δ was the diverger is $\frac{1}{2}$ exactly when the two divergence rates agree. This is McNemar's construction: equal background rates cancel in the contrast. A Wald SPRT tests $\pi = \frac{1}{2}$ against the alternative implied by the shipped pilot band, at $\alpha = \beta = 0.02$, capped at 150 pairs. False positives carry Wald's bound; misses have no corresponding bound. The *unmatched* control is drawn uniformly; the *layer-matched* control matches each member's declared layer.

Table 14: Contrast membership against historical SDD, seed 4. D5 answers violate the benchmark's refusal policy; they need not be wrong causes. Agreement is conditional MCA, and "= cause" counts exact members of the planted minimal-cause family (including either singleton on D2). Executions are workload re-runs per instance; ×SDD compares with 378 / 394 / 1,101 / 2,501 / 3,403.

control	tier	answer rate	agreement when answered	= cause	mean executions	×SDD
unmatched	D3	0.623	1.000	187	6,707	6.1
unmatched	D4	0.217	1.000	65	11,015	4.4
unmatched	D5	0.060	0.944	17	7,700	2.3
layer-matched	D1	0.967	1.000	290	2,256	6.0
layer-matched	D2	0.977	0.997	292	1,468	3.7
layer-matched	D3	0.930	1.000	279	9,648	8.8
layer-matched	D4	0.823	1.000	247	16,876	6.8
layer-matched	D5	0.653	1.000	196	24,050	7.1

Layer matching controls a read-density confound. `ddmin`'s partitions are contiguous index blocks laid out by layer, so a dense block of one layer diverges more than a random set of the same

size. On every failing D3 instance, the first false positive was a 32-factor set that did not activate the cause, at divergence rate about 0.12 against a maximum of 0.096 over random 32-factor sets. The unmatched oracle therefore responds to differences in read density. Matching the control on the declared layer removes that confound: the contrast recovers the injected cause on D5 in 196 of 300 instances with conditional MCA 1.000, at $7.1\times$ the shipped executions and with a coverage loss of 3–17 percentage points on D1–D4. These results locate D5's recovery limitation in the thresholded membership test and quantify the cost of a matched-control alternative. Proposition 3's guarantee concerns the thresholded oracle; a bound for the full contrast procedure is not proved here.

Gate sample-size sensitivity. On seed 4 the acceptance gate refuses 45 of 1,200 D1–D4 instances (0, 4, 2 and 39 by tier), a policy coverage loss of 3.75%. Varying n_{cert} over 30, 60, 120 and 240 (`experiments/gate_ncert.py`) yields 47, 45, 46 and 45 such refusals and 41, 48, 52 and 49 D5 answers. These counts show little sensitivity over the tested range, not that every refused candidate is wrong: Appendix A.3 records refused 1-minimal candidates. A singleton passing all gate comparisons costs $60(1+3+1) = 300$ executions; larger candidates and early rejection change the cost.

C.2 Explain on a real divergence

To test attribution on an observed workload divergence, we intervene on CAMELYON's client-controlled ordering factors using `experiments/sdd_real_divergence.py` ([M], no provider call, run under the CAMELYON environment; pre-declaration in `experiments/results/sdd_real_divergence_predeclared.json`, results in `sdd_real_divergence.json`). The other workloads lack the required intervention evidence. With all seven committed Campaign B caches merged, the unsealed baseline, arrival-order intervention and tie-break intervention cover only 3, 7 and 2 of 30 comparable cases, respectively, leaving their attribution results UNDEFINED (`sdd_real_divergence_campaign_b_coverage.json`). The loop's provider-side residual cannot be held fixed by the client and remains unsplit between sampling and serving effects.

Declaration. The committed §4.1 trace declares one step per slide and puts the arrival-order digest in step 0's arguments, yielding Table 2's first divergence at step 0 and 61 suspects. For finer localization, the attribution experiment declares the seven steps executed by `run_slide` per slide (tissue mask, candidate enumeration, scoring, admission, heatmap accumulation, lesion extraction, size class) and one staging step per patient. The cohort then has 144 steps and 231 declared reads. Tile arrival order and heatmap accumulation order are named dependencies of the admission and accumulation steps, separate from their explicit arguments. This allows comparison of differing outputs under identical explicit inputs while tracking the ordering dependencies. Executions use the tile cache. These attribution recordings have verified chains and no trusted completeness anchors; Appendix B reports the checkpoint status of other CAMELYON artifacts.

Comparator. Two unsealed executions first diverge at step 4, the heatmap accumulation of the first slide, whose read-set has three entries; the union-onward rule returns 227 of 231 reads as suspects because every later slide follows. With the accumulation pinned and admission left in arrival order (a configuration no paper arm has) the first divergent step is 17, the admission of the first binding slide, read-set {candidates, tile budget, that slide's arrival order}, 209 suspects. At the decision level the first slide whose lesion size-class multiset differs is patient 017 node 2 in both pairs. The union rule therefore does not narrow on a cohort of independent slides; the informative localization is the divergent step's own read-set, and the narrowing happens inside the attribution through read-closure pruning and `ddmin`.

SDD. The pre-declared universe has 40 client-controlled factors, per slide the arrival order and the accumulation order with the sealed arm's declared order as baseline. Divergence is measured on the slide's lesion size-class multiset, the level of Table 2's 29 flips, with one execution being a pair of cohort re-runs. 39 factors are suspects, pruning keeps 5, and `ddmin` returns one, the arrival order of patient 017 node 2, in $N = 3$ search membership tests with no test reaching $n_{\max}$ at either historical calibration, at 363 and 375 executions. The products $N\alpha = 0.060$ and approximately 0.0009 are diagnostics, not per-instance risk bounds. The acceptance gate retains the candidate against three

size-matched controls (estimated $p_{\text{pred}} = 1.0$, $p_{\text{ctrl}} = 0.0$). This is a single, empirically well-separated example, not a strict-mode result or evidence for the D4–D5 assumptions.

Table 15: Intervention on the gate-accepted factor, 100 independently randomized cohort pairs per arm. The event is disagreement in the attributed slide's lesion size-class multiset; one-sided 95% Clopper–Pearson bounds concern repeated randomizations of this fixed cohort, not a patient population. Removing the budget yields no observed decision divergence in these pairs despite bit-level accumulation variation.

arm	held at declared order	flips / 100	bound
all perturbed	nothing	100	lower 0.970
cause held	the named arrival order	0	upper 0.030
control held (three arms)	one random other factor each	100, 100, 100	lower 0.970
budget removed, all perturbed	nothing	0 (every slide)	upper 0.030

Intervention. In Table 15, holding the named arrival order fixed removes the attributed size-class divergence and moves the cohort's first divergence to the next slide whose tile budget binds; holding any control leaves the attributed divergence. Node labels flip between micro- and macrometastasis in 20 of 100 all-perturbed pairs (patient 024 node 0). The patient-level nodal stage (pN) flips in 0 of 100 pairs in every arm: all three staged patients have five positive nodes and remain at stage pN2. The staging rule therefore explains the unchanged patient-level result in §4.1. The intervention study costs 1,338 executions and 28,952 slide computations from the cache.

D Full Experimental Tables

D.1 Threshold-margin distribution

The distance from each lesion's largest dimension to the nearest size threshold $\{0.2, 2.0\}$ mm measures how close its class is to changing under a size perturbation. Table 16 reports these margins over the 15 of 20 slides that carry a micron scale.

Table 16: Threshold-margin distribution. Distance from each lesion's largest dimension to the nearest size threshold $\{0.2, 2.0\}$ mm, over the 15 of 20 slides carrying a micron scale.

arm	n lesions	min (mm)	median (mm)	max (mm)
A0 unsealed	1,435	0.0017	0.2311	15.3382
A1 ordered = A2 sealed	1,071	0.0034	0.2311	15.3512
A3 score-ranked declared order	1,439	0.0034	0.2311	15.3512

A fourth arm, and the annotations. A3 declares admission by detector score with a (y, x) tie-break instead of raster position (`configs/arms_a3.yaml`, run from the same cached tile scores, seed 42, 6 repeats, `artifacts/determinism_arms_a3.json`; A0–A2 reproduce Table 2 exactly in the same run). The historical A3 artifact marks all 20 tile caches as weight-digest-unverified: its comparisons hold cached values fixed but do not independently verify their detector provenance. A3 achieves bitwise agreement on 100/100 heatmap and tile comparisons with 0 size-class flips. Its mean admitted-tile score over the 11 binding slides is 0.570 against A1's 0.469 and A0's 0.479, highest on every binding slide. Against the CAMELYON17 annotations for the 9 annotated slides in the cohort (131 annotated components, centroid-in-component matching; `src/sealing/truth_by_arm.py`), A0, A1 and A3 each detect 1 of 131, with 704, 609 and 687 false-positive components and pooled Dice overlap scores of 0.005–0.006: the thresholded heatmap marks 5–6% of pixels against 0.1% truth. All tested orders retain poor lesion recovery and overlap. The detector's missed lesions and false positives therefore limit the clinical interpretation of the reproducibility results in §4.1.

Over the 1,435 unsealed lesions the minimum distance to a threshold is 0.0017 mm and the median 0.2311 mm. These margins identify proximity to a size boundary; actual crossings depend on the perturbation. Appendix C.2 measures node-label flips and explains why every staged patient remains

pN2 with five positive nodes. The lesion count n also changes between arms because unsealed runs produce more connected components.

D.2 Float-associativity probe

To isolate floating-point accumulation effects, we permute only accumulation order while holding detector outputs fixed. This moves a tile probability by one float32 unit in the last place (ULP; 5.96×10^{-8}) over an 81-window probe and two ULP (1.19×10^{-7}) over the full 7,679-tile slide. In both probes, 0 of 8 permutations achieve bitwise agreement with the reference. A compensated summation path is permutation-invariant in these probes and achieves bitwise agreement with `math.fsum`. The companion admission probe measures a different ordering effect: 1,316 of the 6,000 retained tiles are present in one run only.

D.3 Propagation from ordering to lesion size class

Figure 3 follows ordering-induced differences through heatmap construction, lesion extraction and size classification to show where numerical changes become decision changes.

D.4 Expanded CAMELYON cohort

The frozen extension contains 100 slides from 20 challenge patient groups, four groups per center and five slides per group. It retains the original four groups as the ordered prefix and adds 16 groups selected by a fixed hash ranking within centers. These are artificial challenge groupings, not verified independent biological patients. The detector digest `472c3e06...`, level 4, 64-pixel patches, stride 32, 6,000-tile budget and six-run seeds are unchanged. The reproduced baseline and expansion share one recorded environment fingerprint; the remote execution report records Python 3.13.11, torch 2.11.0+cu128, CUDA 12.8 and cuDNN 91900 on an L40S, differing from the historical stack. All specified original-cohort measurements reproduce exactly, and the original prefix's raw records match inside the expansion.

Table 17: Expanded 100-slide numerical validation. Heatmap and tile rows compare runs 1–5 with run 0. M4 uses a separate per-slide pair. Matched-lesion denominators differ by arm and exclude slides without a micron scale. The stage-pair row excludes 20 both-refused pairs per arm; no availability transitions occur. All arms have 80/100 slides with a micron scale and 55/100 with a binding budget.

	A0 unsealed	**A1 ordered**	**A2 ordered**
heatmap agreement	0/500	500/500	500/500
tile-selection agreement	225/500	500/500	500/500
matched-lesion class changes (M4)	203/5,434	0/5,396	0/5,396
consecutive group-stage flips	0/80	0/80	0/80
group-stage changes (M4)	0/16	0/16	0/16
refused challenge groups	4/20	4/20	4/20

The added 80 slides contribute 400 reference comparisons per arm and 174 unsealed M4 class changes; both ordered arms have complete heatmap/tile agreement and zero M4 class changes. The expanded A0 cascade has 68,872,928 threshold-crossing pixels, an absolute component-count-difference sum of 241, and maximum matched-lesion axis difference 2.4539 mm. Unmatched components are excluded from the class-change count. The extension measures numerical ordering effects only; replay, attribution and annotation accuracy retain their original-cohort scope.

Evidence. `experiments/camelyon_cohort_expansion.py` produces the summaries and raw records. Under `examples/camelyon/artifacts/`, the reproduced baseline is `original20_seed42_20260914T171005Z_46e8a23f/` and the expansion is `cohort_l4_100_seed42_debug_20260914T211200Z/`. The numerical supplement includes their manifests, completion hashes, cache inventories, summaries and raw six-run/M4 records. Checks recompute outcomes and verify all distributed completion-bound files and prefix equality. Cache inventories record 100 verified identities: 20 baseline caches reused and 80

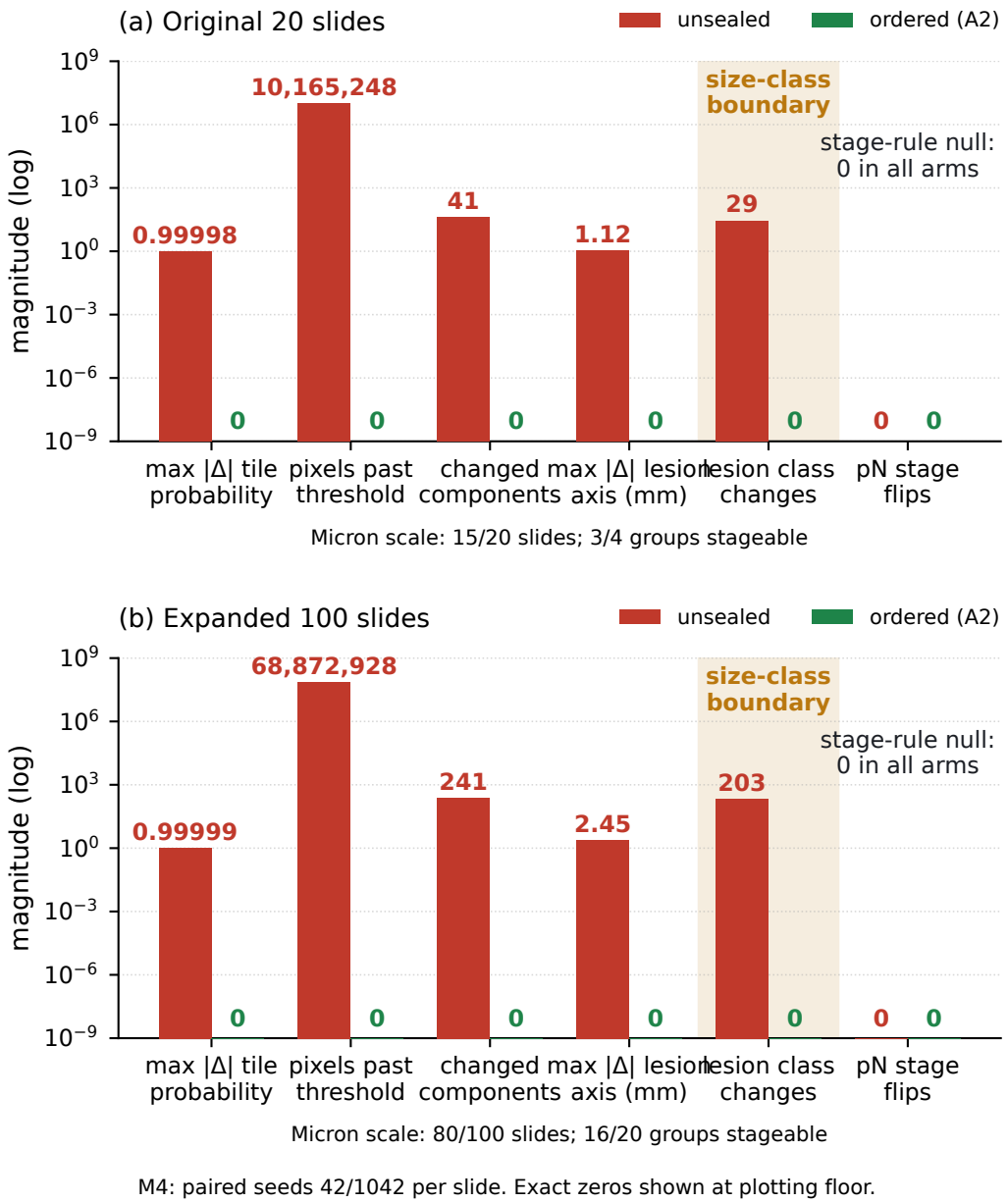

Figure 3: Ordering-induced differences in the original 20-slide cohort and nested 100-slide extension (Appendix D.4). Each panel uses the separate M4 two-run protocol, resetting seeds to 42 and 1042 per slide. Each stage has its own unit; counts also depend on cohort size. Bars are log-scaled and exact zeros are marked at the floor. A1 ordered-only also has zero differences throughout both cascades. Lesion-class changes count matched lesions on slides with a micron scale. Group-stage changes are zero among 3 and 16 stageable challenge groups, respectively, in every arm; this is a staging-rule null, not an ordering benefit or a clinical-accuracy result.

newly scored. Checkpoint/cache bytes, private environment records and the external audit file are not distributed; the public checks establish numerical-record consistency, not independent image re-extraction or execution authenticity.

D.5 THE [M] TOOL-LOOP CONTROL, 30 DRAWS

This control replaces the model with a fixed first-candidate selector, isolating the client-side race as the source of divergence. Each unsealed cell reports the median over 30 independent draws followed by that quantity's [min, max]. In the range row, the minimum was 2 in every draw; the per-draw maximum had median 8 and ranged over [7, 13]. Every sealed cell was identical in all 30 draws. L_{tool} counts tool attempts including retries, with a cap of $8 \times 3 = 24$ from the call and retry budgets in Appendix B.1. This arm has 2,160 episodes over 30 draws, versus the [R] arm's 288 over four. Its larger pooled maximum therefore covers more episodes; its median per-draw maximum is 8 versus the billed campaign's observed maximum of 9.

Table 18: The [M] tool-loop control, 30 draws. The identical workload with the model removed, so the client-side race is the only source of divergence. Each unsealed cell gives the median over 30 executions followed by that quantity's [min, max]; every sealed cell was identical in all 30.

	unsealed (30 draws)	**ordered + SEAM**
cases whose L_{tool} varied	6 / 6 in every draw	0 / 6 in every draw
L_{tool} observed range	2 – 8 [7, 13]	2 – 5, constant
distinct tool sequences / 72 episodes	69.5 [67, 71]	6, constant (the floor)
cases whose decision varied	4 / 6 [2, 5]	0 / 6 in every draw
retries; tool executions	53.5 [37, 73]; 307 [280, 332]	24; 276, both constant

Per case, the [R] campaign's unsealed L_{tool} runs 2–7, 3–7, 2–7, 3–9, 3–7 and 3–7, so all six cases vary, while no single case spans the pooled 2–9 range.

The fixed selector reaches the sealed tool-sequence floor, one distinct sequence per case, in all 30 draws. Together with differing provider responses to identical recorded prompts, this locates a remaining source at the provider boundary without separating sampling from serving numerics. A fixed graph can also propagate early stochastic outputs into later prompts, as Campaign B's specialist-to-critic edge does. The loop additionally lets those outputs choose control flow and tool-attempt count.

An open-weights model in the loop, greedy and sampled. [M], from `experiments/toolloop_local.py` over six artifacts produced by `toolloop.py --mode local`: the same six cases, twelve repeats, two arms, prompt bytes, 256-token cap and answer parsing as the billed campaign, with the provider replaced by `Qwen/Qwen2.5-14B-Instruct` on one L40S in bfloat16 (bf16) precision, batch size 1, deterministic algorithms enforced, revision pinned in the artifact. Greedy decode is `do_sample=False`; the sampled arm uses temperature 1.0 with a fresh unpredictable seed per call, recorded. Greedy draws 1 and 2 and one sampled draw each use 549–564 model calls and take about 270 s.

Table 19: The loop on a locally served open-weights model. Execution metrics use the sealed arm except the last row. Within-prompt choice disagreement compares normalized next-step choices for pairs of calls sharing identical prompts, using each recording's cache across both arms. The [R] column summarizes the provider campaign's four default recordings.

	greedy 1	**greedy 2**	**sampled**	**[R], 4 draws**
distinct tool sequences / episodes completed	6 / 72	6 / 72	18 / 72	12–18 / 70–72
cases varying tool sequence / L_{tool} / decision	0 / 0 / 0	0 / 0 / 0	6 / 5 / 1	5–6 / 2–4 / 0–1
within-prompt choice disagreement	0 / 1,481	0 / 1,485	203 / 1,259	0.1106 pooled
tool-sequence replay; internal trace check	yes	yes	yes	3 of 4 default draws
unsealed distinct tool sequences / completed	72 / 72	70 / 72	72 / 72	69–71 / 72

Both greedy draws have identical sealed summaries and per-case statistics, and the 70 prompts they share receive identical normalized next-step choices. The observation establishes choice-level agreement under greedy decoding on these prompts. Enabling sampling on the same local model introduces tool-sequence, tool-attempt-count and decision variation, isolating a sampling contribution under this controlled configuration. Identifying the provider-side mechanism or its magnitude would

require comparable control over the provider's sampling and serving configuration. A 7B model of the same family returns a query outside the offered window in about a fifth of decisions, causing 60 of 72 sealed episodes to end unusable in every draw. This is a prompt-compatibility limitation, comparable to the provider-model failures in Appendix D.7. Its one answerable sealed case has a single tool sequence under greedy decoding and varies when sampled.

D.6 Provider choices on identical prompts

Retrospective producer `experiments/measured_sampling_hazard.py` compares normalized next-step choices for pairs of provider calls with byte-identical prompts in the four default loop recordings. Disagreements are 116/1,375 in draw 1, 192/1,344 in draw 2, 130/1,382 in draw 3 and 165/1,351 in draw 4: rates of 0.0844, 0.1429, 0.0941 and 0.1221, pooled 0.1106 over 5,452 pairs. These normalized choice pairs measure provider-side variation on the visited prompts; they neither establish independent calls nor isolate sampling from serving effects.

D.7 Tool-loop configuration and temporal replication

A draw with the reasoning block disabled. The loop's model accepts `thinking: disabled` (Appendix B), so one further draw (504 model calls, 2026-09-05, `experiments/toolloop_nothink.py`) ran both arms with it. Every response is a bare text block of exactly 10 output tokens and none is unusable (the four default draws have 0–2). The unsealed arm is unchanged (70 distinct tool sequences over 72, 6 of 6 cases varying, 4 of 6 decisions varying). The sealed arm moves toward the floor without reaching it: 8 distinct tool sequences over 72 against the default draws' 12–18 and the floor of 6, 2 of 6 cases varying tool sequence against 5–6, 0 varying decision, and identical-prompt disagreement 31 of 1,427 pairs (0.022) against 0.084–0.143. This configuration has lower observed disagreement, but disabling reasoning changes the execution path and downstream prompts. It does not decompose the residual into reasoning and answer-token sampling, nor rule out serving effects. The draw is not pooled with the four default-decode draws.

Temporal replication at default decoding. A fifth draw uses the same configuration 1.9 to 3.9 days after the first four (515 billed calls; `experiments/toolloop_epoch2.py`). This interval is shorter than the approximately 20 days between panel epochs, so the two replications test different temporal horizons. Of 22 summary quantities, 21 of 22 lie within the original observed range, including 9 of 9 previously constant quantities. These ranges summarize the recorded evidence without providing population bounds.

The fifth draw retains byte-identical initial inputs in 6 of 6 sealed cases, varying tool sequences in 5 of 6, and 12 distinct sequences against a floor of 6. Tool replay and internal trace checks pass. Only the unsealed retry count lies outside the original range: 61 against 48–60. This recording is reported separately to preserve the temporal comparison.

Cross-family prompt compatibility. Table 8's model probe records `opus` and `fable` returning empty answers to this prompt at rates 1.00 and 0.75, against sonnet's 0.00. A campaign-scale draw on `opus` produces 72 of 72 unusable episodes in both arms, each failure containing empty text at 1–3 output tokens below the output cap. The response records therefore show prompt-compatibility failures without cap exhaustion. These failures prevent a usable cross-family provider comparison of output propagation. Appendix D.5 supplies an open-weights comparison; generality across provider models remains untested on usable responses.

E Deterministic Replay: Related Systems

SEAM instantiates established record-replay and integrity mechanisms at tool and model boundaries. Instant Replay records shared-access order (LeBlanc & Mellor-Crummey, 1987); ReVirt and `rr` extend record-replay to whole-system and practical debugging settings (Dunlap et al., 2002; O'Callahan et al., 2017). Content addressing and authenticated histories likewise predate SEAM (Merkle, 1988; Haber & Stornetta, 1991; Crosby & Wallach, 2009). Our signature over trace length and terminal digest protects an already anchored recording from subsequent modification, subject to

the trust assumptions in Appendix B. None of these primitives is claimed as novel. The comparison below tests their configured behavior at agent boundaries.

Comparator read-set (§4.1). The comparator returns the union of declared reads over every step from the first divergence onward, providing a dependency-based starting point for attribution. In the unsealed CAMELYON arm, divergence occurs at step 0, so the 61 suspects comprise the entire declared read set. Complete declarations require all relevant state reads to pass through the instrumented boundary and every external or implicit dependency to be named. Listing input-state keys misses a dependency if the node also reads an undeclared file, clock, global variable or service. The suffix union can exclude earlier-only reads when traces initially agree, including earlier inputs that cause delayed divergence. It therefore supplies suspects without guaranteeing a complete causal explanation. The sealed arm's empty read-set records the absence of a divergent step.

A tighter read-set would need key-level access tracking inside the node (recording which state keys it dereferences) or a static over-approximation from the node's code; the adapter does neither, and missing a read would make even that dependency record incomplete.

E.1 Measured against HTTP-cassette replay

We compare SEAM with HTTP-cassette replay (`vcrpy`), which records HTTP requests and responses for later reuse, on the same recorded call sequences. The producer, `experiments/cassette_baseline.py`, uses no provider calls [M]. Table 20 measures replay fidelity, refusal of requests that depart from the recording, mutation detection and step localization. Replay fidelity, axis (a), is reported over all four draws. Axes (b)–(d) use draw 2's sealed arm (247 model calls); the measured outcomes are identical across all four draws. Before scoring, the producer checks that each draw's recorded sequence reconstructs its billed-run summaries.

We test default and body-matched cassette configurations. The default `match_on` excludes the request body. Every model call in this workload is a POST to one URL differing only in its JSON body, so all 247 scored calls match one key under the default configuration.

The last column tests a constructed extension: a 68-line sidecar hash chain over the cassette's serialized interactions, verified before request matching. This ablation separates the benefits of integrity checking from those of step and dependency declarations.

Table 20: SEAM against HTTP-cassette replay on four axes. Axes (b)–(d) are scored on draw 2's sealed arm, 247 model calls; axis (a) is measured over calls in all four draws. SEAM v1 (frozen) is the historical last-response reader, included to measure the effect of occurrence-aware lookup on the same recordings. The last column adds a constructed sidecar hash chain to the cassette and verifies it before matching.

axis	SEAM	SEAM v1 (frozen)	cassette, body	cassette, default	+ chain
(a) calls replayed with bitwise agreement	1.000	0.891–0.921	1.000	1.000	1.000
(b) diverging driver refused	1.000, at the diverging call	1.000 diverging, 0.000 over-consumption	1.000	0.000	1.000
(c) single-bit record mutation detected	1.000, naming the first bad line	1.000	0.000	0.000	1.000
(d) divergence localized to a step	1.000; suspect set = all 5 declared factors	1.000	0.000	0.000	0.000

Frozen-reader defect. The frozen reader retains only the last response for each `(tool, args)` request. Earlier occurrences with different responses replay incorrectly: 220 of 247 calls have bitwise agreement in draw 2, with fidelity 0.921 / 0.891 / 0.916 / 0.918 in draws 1–4 (20 to 27 wrong calls). Both cassette configurations achieve 1.000 in every draw by consuming the first unplayed match. They refuse an extra call after recordings are exhausted; the frozen reader reuses the last response indefinitely.

Current occurrence lookup. The index holds each request key's recorded occurrences in order, and replay consumes them in order. Current SEAM achieves 1.000 on all four draws and distinguishes missing requests from exhausted recordings (`ToolReplayExhausted`). The 25-line

reader change uses occurrences already retained by the append-only index, requiring no re-recording or added occurrence fields in typed tool arguments. The frozen and current readers score 0.891 and 1.000 on the same draw-2 bytes, isolating the effect of lookup behavior.

Separate unexplained failure. Axis (a) tests the model boundary. The historical draw-4 failure occurred during tool replay, where no request key repeats within an episode (Appendix B). The occurrence-reader defect therefore does not explain it. Tool replay succeeded in 3 of the 4 default-decode draws, and draw 4's original failure remains unattributed.

Frozen-reader error rate. Draw 4 has 104 calls on 11 requests receiving multiple distinct responses, versus draw 2's 97 calls on 9 requests, yet its replay fidelity is higher (0.918 versus 0.891). Fidelity depends on the frequency of the particular response retained last, not just how many calls share a variable-response key. A 6/6 split loses six matches; an 11/1 split loses one or eleven, depending on which response was last. Draw 2 is the least favorable observed draw for the frozen reader.

Corpus consistency check. Driving `run_loop` from a committed occurrence-indexed cache reproduces all 7 sealed-arm summary statistics of all 4 billed draws. The script refuses measurement if any draw disagrees, checking that the baseline comparison preserves the recorded campaign summaries.

Request matching, axis (b). Excluding the body makes a diverged request match the same key and receive the next response in file order, giving a detection rate of 0.000. Body matching reaches 1.000, equal to SEAM. The result identifies a configuration requirement: the matcher must include request fields that distinguish the workload's calls.

Axes (c) and (d) are absent from the tested unextended cassette configurations. The body-matched cassette's 64 mutation outcomes partition into 20 replay misses, 11 mutated responses served, 14 with no replay effect, 9 load errors, 1 matcher crash and 9 silently dropped interactions. None is an explicit integrity diagnosis. SEAM detects 64 of 64 isolated bit mutations and names the first bad line; this does not cover an adversary rewriting and rehashing the record. An unmodified-record claim needs a trusted digest or checkpoint, and detecting a valid truncated prefix needs the trusted length/terminal anchor. On (d), the default request matcher has only one key, while body-matched request streams never differ in 5 of 6 divergent pairs. A response-side diff can locate the interaction in all six, but does not itself supply step alignment or declared dependencies.

Integrity extension, axes (c) and (d). Our constructed 68-line hash-chain ablation detects 64 of 64 mutations and names the affected interaction, matching SEAM on axis (c). Verification before request matching reduces mutations reported as replay misses from 20 to 0 by diagnosing record corruption directly. Step localization, axis (d), remains at 0.000. The artifact records three missing elements: a cross-run comparator that accepts two recordings, a per-step index that maps interactions to execution steps, and declared read-sets that name step dependencies. This extension tests integrity alone; a declaration-aware extension would be needed to evaluate the same localization interface.

Truncation caveat. The cassette sidecar also detects the tested tail deletion without a signed count, but through serialization framing rather than authenticated completeness. Its final hashed interaction span includes the YAML document trailer; deleting the interaction changes that span. By contrast, deleting complete JSONL trace lines leaves a valid chain prefix. Catching this particular cassette deletion therefore does not establish general completeness without a trusted anchor.

HTTP observation boundary. This draw makes 247 model calls accessible to an HTTP interceptor and 271 in-process tool executions, of which 0 use HTTP. The unsealed arm produces 71 distinct tool sequences over 72 episodes through an `as_completed` race over in-process tasks. An HTTP cassette can preserve the model responses but leaves the schedule generating those 71 sequences uncontrolled. CAMELYON's data reads do use HTTP; its variation arises when arrival order determines admission under a tile budget. Preserving response bytes therefore needs to be combined with control over their consumption order to reproduce that computation.

Integration scope. The loop invokes `SealedGraph` tool nodes directly from a Python loop, with recorded model choices supplied by `_FixedChoices` during tool replay; it does not compile a LangGraph workflow. Campaign B's medical panel does compile a LangGraph workflow and wraps it in `SealedApp`. CAMELYON's sealing probe uses the runner-based path; these are not three independent evaluations of a compiled LangGraph integration.

Comparison scope. This mechanism comparison uses recorded evidence. The tested cassette configurations match SEAM on occurrence-aware replay, refusal of differing requests when body matching is enabled, and isolated-mutation detection with the sidecar chain. The step/read-set comparison uses unequal observability because the cassette lacks SEAM's declarations. An end-to-end cassette deployment, a declaration-aware event log or workflow checkpointer at parity, and relative debugging cost remain unevaluated. The CAMELYON union includes all 61 declared reads, and the six cassette comparisons return the same five factors, so the measured localization results do not establish useful suspect-set reduction or a capability unique to SEAM.

F MECHANIZATION ABLATION

Proposition 2 establishes that a deterministic replacement can increase disagreement; Campaign B evaluates replacement subsets and allocation choices on recorded workflow evidence.

F.1 MECHANIZE ON CAMPAIGN B

Retrieval-ranked verifier set. To measure replacement effects in a model-driven workflow, we use Campaign B's three model roles: two specialists that propose diagnoses and a critic whose prompt includes their adjudicated leading diagnosis. Retrieval windows are fixed across the 30 cases $\times$ 8 repeats, but critic prompts can vary with specialist votes. A plurality rule with canonical tie-breaking produces the final label. We offer a role-specific deterministic replacement: V_{spec} for either specialist and V_{critic} for the critic. This gives 8 replacement subsets S at $3 - |S|$ model calls per run (`experiments/mechanize_campaign_b_offline.py` and `mechanize_campaign_b.py`).

V_{spec} returns the highest-ranked knowledge-base document's label. V_{critic} confirms specialist unanimity; otherwise it selects the split label whose document appears earliest, ranking absent labels last and breaking remaining ties alphabetically. Both cost no model calls [M], but the critic replacement can still vary with specialist inputs. V_{spec} agrees with the modal generalist label on 0 of 30 cases and with gold on 2 of 30; V_{critic} agrees with the recorded critic on 200 of 240 repeats.

Prediction from the baseline recording. The roles are not independent: the critic's prompt is a function of the leader, so per case the decision distribution is computed exactly from the specialists' joint label distribution and the critic's distribution conditional on that leader. The calculation is exact for the fitted empirical distributions; population prediction additionally depends on their estimation accuracy. A specialist replaced by V_{spec} becomes a point mass and the other keeps its marginal, since its prompt is unchanged. Where a mechanized specialist implies a leader the case's critic never saw, a pooled fallback over the verifier set's split-panel critic calls is substituted and flagged (2 case-configuration pairs). The offline artifact, which reads no mechanized recording, fixes the predictions for all 8 configurations: it ranks `spec_id` (0.023) and `spec_gen` (0.032) below the live panel (0.035) and finds 6 purchases in 4 cases that raise predicted PDR, every one of the form "replace one specialist". A prediction that treats the three roles as independent is worse wherever it differs (mean absolute error 0.040 against 0.025 on `spec_gen`, 0.026 against 0.019 on `spec_id`).

Prediction provenance. The offline artifact records the epoch-1 recording's SHA-256, and the realization script recomputes identical predictions from it. These checks establish content provenance and deterministic computation, not temporal ordering. Pre-registration chronology is author-reported; no independently timestamped commitment is supplied.

Realization. The three configurations retaining a live call were recorded on 2026-09-06 on the same model with windows byte-identical to epoch 1's: 1,200 billed calls, 0 failures. All 480 live-specialist prompts match prompts sent in epoch 1. Critic prompts match epoch-1 prompts in 232 of 240 runs of each single-specialist configuration and in 0 of 240 runs when both specialists are verifiers. The three configurations that add V_{critic} are re-decided from this recorded evidence without further calls.

What matched and what did not. Prediction and realization agree exactly on six configurations: the two realizable from epoch 1 itself and the four in which two verifier votes fix the decision, so PDR is 0 by construction. The two single-specialist configurations were under-predicted, 0.032 against 0.055 and 0.023 against 0.040, and the realized order puts both above the live panel where the prediction put them below. Re-predicting with the new recordings' rates swapped in one component

Table 21: Mechanize on Campaign B, epoch-1 sealed verifier set, all 8 configurations. Predicted PDR is pre-registered from epoch-1 rates; realized is the mean plug-in over 30 cases with the disagreeing pairs of 840; gold agreement is the fraction of the 240 decisions equal to the case label. Configurations with a live call were billed; those adding V_{critic} are re-decided from them.

mechanized roles	calls/run	predicted	realized	pairs	gold	class
none (live panel)	3	0.035	0.035	34/840	0.200	[R]
critic	2	0.035	0.035	34/840	0.167	[R]+[M]
generalist	2	0.032	0.055	53/840	0.192	[R]
infectious disease	2	0.023	0.040	38/840	0.204	[R]
both specialists	1	0.000	0.000	0/840	0.067	[R]
generalist + critic	1	0.000	0.000	0/840	0.067	[R]+[M]
infectious disease + critic	1	0.000	0.000	0/840	0.067	[R]+[M]
all three	0	0.000	0.000	0/840	0.067	[M]

at a time attributes the gap to the critic's eight fresh draws on byte-identical prompts (+0.017 and +0.018), not to the fallback (-0.001, -0.009) or to the live specialist (+0.004, +0.000); the critic's pooled confirm-the-leader rate barely moves, 192 of 240 to 196 of 232, so sparse per-case estimation is a plausible source of error. Similar aggregate critic rates do not rule out case-specific or temporal drift.

Predicted increases. All 6 of 6 pre-registered disagreement-increasing purchases also increased the observed estimate, and one additional increase appeared where prediction gave a tie. On consecutive pairs `spec_gen` flips 17 of 210 (nominal pooled CP upper limit 0.1190), above $\varepsilon = 0.1$ where the live panel's nominal limit is below it; neither is a dependence-valid certificate, and the instability moves across strata, up in the afebrile stratum (7 to 37 of 560 pairs) and down in the hypoxic one (27 to 16 of 112).

Allocation policies. We compare four policies, both pooled and per case, at budgets of 1, 2 and 3 replacements. At budget 1, budget-filling and role-instability ranking replace a specialist in 28 and 25 of 30 cases. Greedy strict improvement buys only in the 3 cases whose live panel varies, retaining the other 27; its observed PDR is 0 at 2.90 calls per run and gold agreement 0.200.

Among 87 (case, budget) purchases declined by greedy selection but made by budget filling, 86 give the same observed PDR, 1 worsens it, and 0 improve it. The worsening purchase has predicted gain zero, observed PDR 0.219, and gold agreement falling from 1.0 to 0.125. At the pooled level, greedy selection buys `spec_id` on a predicted decrease that becomes an observed increase, showing how prediction error can reverse the allocator's expected benefit. Full replacement is worse than the best subset in 0 of 30 cases on PDR and 7 of 30 on gold agreement: verifier votes fix the decision and eliminate disagreement while lowering gold agreement from 0.200 to 0.067.

Signature-based verifier set (V_{sig}). The retrieval-ranked specialist verifier agrees with the panel's modal label on 0 of 30 cases. To examine a replacement that also preserves some correct panel decisions, we evaluate a signature-based verifier (`experiments/mechanize_campaign_b_v2_offline.py`, `mechanize_campaign_b_v2.py`; recordings `medical_demo_evidence_mech_v2_spec_gen.json` and `medical_demo_evidence_mech_v2_spec_id.json`, 960 billed calls on the same model, 2026-09-07, 0 failures). Its lookup table uses a coarse presentation signature consisting of test-panel size and six vital thresholds. It is fitted on synthetic labels from the 4,970 corpus rows outside the 30 evaluated cases, reads only prompt text, and costs no provider call. It agrees with gold on 17 of 30 cases against the panel's 6, and with the panel on 5 of the 6 cases where the panel is correct. Each recording carries the fitted table.

As with the retrieval-ranked verifier set, pre-registration is author-reported without an independently timestamped commitment. The both-specialists configuration is flagged as realized by construction because two verifier votes fix the decision, removing the need for a billed run. Here V_{sig} returns the same signature-table label for each replaced role, including the critic; the retrieval-ranked verifier set uses a panel-dependent critic rule. Mechanized recordings were collected after the baseline epoch, leaving temporal effects unresolved in billed single-specialist contrasts. The critic-only contrast re-decides the same recorded specialist votes.

Table 22: Mechanize on Campaign B under the signature-based verifier set V_{sig}, all 8 configurations; columns as in Table 21. Under the retrieval-ranked verifier set V_{spec}, every configuration with two verifier votes had gold agreement 0.067.

mechanized roles	calls/run	predicted	realized	pairs	gold	class
none (live panel)	3	0.035	0.035	34/840	0.200	[R] recorded-evidence reanalysis
critic	2	0.029	0.029	28/840	0.208	[R] recorded-evidence reanalysis + [M]
generalist	2	0.131	0.065	62/840	0.229	[R] billed
infectious disease	2	0.113	0.046	44/840	0.237	[R] billed
both specialists	1	0.000	0.000	0/840	0.567	[M] by construction
generalist + critic	1	0.000	0.000	0/840	0.567	[R] billed + [M]
infectious disease + critic	1	0.000	0.000	0/840	0.567	[R] billed + [M]
all three	0	0.000	0.000	0/840	0.567	[M] by construction

Mechanizing the critic lowers PDR from 0.035 to 0.029 while gold agreement rises from 0.200 to 0.208, and every configuration with two verifier votes reaches 0 at gold agreement 0.567; the corresponding nominal CP limits are 0.0142, 0.0247 and 0.0950 (0/210 consecutive pairs, 0/120 disjoint pairs, 0/30 cases). Pair limits do not account for dependence, and the case limit needs iid case events. The zeros follow from the voting construction and are not independent validation of the allocator.

Single replacements. The single-specialist configurations raise PDR (0.065 and 0.046 against 0.035) while raising gold agreement (0.229 and 0.237 against 0.200), the opposite pairing from the retrieval-ranked verifier set: a lone verifier that dissents from a unanimous panel hands the decision to the critic, which sided with the live specialist on 44 of 55 and 44 of 56 such prompts, so the decision usually stayed the model's and where it did not the rate rose. The pre-registration ordered all 8 configurations correctly (0 inversions of 28 pairs) and identified every replacement that raised observed disagreement (8 realized of 22 predicted increases), but over-predicted the two single-specialist rates (mean absolute error 0.083 and 0.074) because its pooled fallback assumed the critic would confirm a verifier-only leader 16% of the time and it did so on 2 of 55 prompts.

Greedy versus joint selection. Pooled, all four allocator policies buy the critic at budget 1 and the strict-improvement rule reaches a two-verifier configuration at budget 2. Per case, greedy selection retains the live panel in 28 of 30 cases. Of 88 declined purchases, 85 tie the retained PDR, 1 would worsen it, and 2 would improve it. Both missed gains concern one case: no single replacement improves predicted PDR (0.469 → 0.594, 0.469, 0.656), but any pair reaches 0 with gold agreement 1.0. Joint selection finds this pair at budget 2; greedy selection stops before it. The pinned replication shows the critic purchase raising PDR where epochs 1 and 2 show it falling, so that improvement is not robust across recordings.

Estimator and uncertainty. For m labels with counts n_l, the reported plug-in is

$$\widehat{\mathrm{PDR}}_{\mathrm{plug}} = 1 - \sum_l (n_l/m)^2 = \frac{m-1}{m} \frac{\sum_{r<s} \mathbf{1}\{Y_r \neq Y_s\}}{\binom{m}{2}}.$$

At $m = 8$, the 34/840 pair fraction is 0.04048 and its mean plug-in is 0.03542. The pair fraction is unbiased under iid repeats within a case; the plug-in has finite-repeat downward bias. Table 23 reports paired case differences from `experiments/results/mechanize_uncertainty.json`, with all repeats and both configurations kept together in each of 20,000 case-bootstrap resamples (seed 20260909).

The joint rule minimizes predicted PDR, then verifier count, then configuration name. Historical exact-search comparisons also used predicted gold agreement as a tie-break; the retrospective gold-blind reanalysis in Table 23 gives the same aggregate contrasts. The critic changes PDR on only two cases per recording and gold agreement on one. Intervals condition on recorded evidence, fitted verifiers, predictions and choices, omitting provider-epoch, verifier-fit and selection uncertainty. Sparse changes and zero-width intervals cannot establish equality or accuracy non-inferiority. The pinned sign reversal also prevents a recording-independent improvement claim. Agreement with gold is a separate measured outcome: V_{sig} reaches 0.567 against synthetic labels when verifier votes determine the decision. Clinical utility and population safety require evidence beyond these synthetic-label comparisons.

Table 23: Recording-conditional mechanization differences, estimate [approximate 95% percentile case-bootstrap CI], 30 cases. Critic rows are V_{sig} critic minus the same recording's live panel. Allocator rows are greedy strict improvement minus exact joint selection at the same installation budget, using predicted PDR without gold labels.

comparison	$\Delta\widehat{\mathrm{PDR}}_{\mathrm{plug}}$	Δ gold agreement
critic, epoch 1	-0.00625 $[-0.03750, 0.01875]$	0.00833 $[0, 0.02500]$
critic, epoch 2	-0.00729 $[-0.03750, 0.01563]$	0.02083 $[0, 0.06250]$
critic, pinned	0.02083 $[0, 0.05729]$	0.00417 $[0, 0.01250]$
V_{sig}, greedy–joint, budget 1	0 $[0, 0]$	0 $[0, 0]$
V_{sig}, greedy–joint, budgets 2–3	0.01563 $[0, 0.04688]$	-0.03333 $[-0.10000, 0]$
V_{spec}, greedy–joint, budgets 1–3	0 $[0, 0]$	0 $[0, 0]$

G CAMPAIGN REANALYSIS

Case-level campaign reanalysis supports the current sampling audit in §4.3 by separating pair-risk and repeat-set targets and examining within-case dependence, strata and recording epochs.

Historical calibration status. Older synthetic calibration recordings used mismatched population and score-label designs: calibration and evaluation drew different shared predicate offsets, and repeat-3 scores were paired with labels from unrelated executions. Those recordings remain in the artifacts and do not validate §3.3's current scan.

Tamper localization over the attribution corpus. Five committed seed runs detect 726 of 726 tamper events, with 0 mislocalized. Of these, 265 are CAS corruptions without a line label, so they are scored only for detection. The remaining 461 have a labeled location (447 a line, 14 a replay step), which the detector identifies exactly. This separates detection coverage from the subset on which localization is measured.

Table 24: Per-stratum one-sided 95% CP limits, 30-case campaign. Pair rows target pair disagreement and are nominal because within-case dependence is not modeled. Case rows target any disagreement across eight repeats and require iid case events for binomial validity. "Below" is numerical comparison to $\varepsilon = 0.1$, not a certificate for nominal rows.

unit	stratum	unsealed	sealed	pinned	sealed limit below ε?
consecutive repeat pairs					
	afebrile (20)	14/140 → 0.1519	2/140 → 0.0443	6/140 → 0.0828	yes
	febrile (6)	10/42 → 0.3703	0/42 → 0.0688	0/42 → 0.0688	yes
	hypoxic (4)	1/28 → 0.1585	6/28 → 0.3797	0/28 → 0.1015	no
disjoint repeat pairs — the historical nominal calibration unit					
	afebrile	9/80 → 0.1881	1/80 → 0.0579	3/80 → 0.0941	yes
	febrile	6/24 → 0.4347	0/24 → 0.1173	0/24 → 0.1173	no
	hypoxic	1/16 → 0.2640	3/16 → 0.4166	0/16 → 0.1707	no
case-level cluster					
	afebrile	5/20 → 0.4556	1/20 → 0.2161	2/20 → 0.2826	no
	febrile	3/6 → 0.8468	0/6 → 0.3930	0/6 → 0.3930	no
	hypoxic	1/4 → 0.7514	2/4 → 0.9024	0/4 → 0.5271	no
sealed strata clearing ε: 2 of 3 consecutive, 1 of 3 disjoint, 0 of 3 case-level					
best possible case-level bound at zero flips: 0.1391 (afebrile), 0.3930 (febrile), 0.5271 (hypoxic)					

The febrile nominal limit crosses 0.1 between consecutive and disjoint pairs, without either becoming dependence-valid. The four-case hypoxic stratum has more observed disagreement under sealing. Within each stratum, even zero case-any events would leave the iid-binomial upper limit above 0.1 at these sample sizes: that target needs at least 29 independent cases per stratum at zero events. This is not a statement about pooling all 30 cases, where zero events would give 0.0950.

Location of sealed-arm disagreement. In the same recording, `experiments/cascade_strata.py` finds that all 3 sealed cases with decision flips have one retrieved-evidence window across their 8 repeats, while specialist votes vary on byte-identical prompts. These include 2 of the 4 hypoxic cases, accounting for that stratum's higher

observed disagreement under sealing. In 2 of the 3 cases, a changed specialist vote changes the adjudicated leading diagnosis and hence the critic's prompt. This produces the 92 distinct model inputs against a floor of 90 reported in §4.3. Across the 240 sealed runs, 11 depart from their case's modal specialist-vote pattern, 4 change the leader, and 6 change the decision.

Per-call disagreement on byte-identical input. From `experiments/campaign_b_identical_inputs.py`: grouping calls by exact prompt bytes, the sealed arm's specialists return different labels on 65 of 1,680 same-prompt pairs (0.039; 16 of 420 consecutive pairs) and the critic on 64 of 818 (0.078) in epoch 1, 0.041 and 0.070 in epoch 2, and 0.050 and 0.066 over the 90-case campaign, giving approximately 4–8% same-prompt pair disagreement in these recordings. In epoch 1 it is larger for the critic than for either specialist (generalist 0.026, infectious-disease 0.051). The unsealed arm repeats a prompt too rarely to measure it (424 distinct specialist prompts for 480 calls).

Window content. From `experiments/campaign_b_window_lead.py`: the sealed window is content-ranked, and because every presentation mentions oxygen saturation, one oxygen document leads it in 176 of 240 sealed runs against 4 of 240 unsealed (408 of 720 against 13 of 719 at 90 cases). Sealed runs led by that document split the two specialists in 58 of 408 against 4 of 264 led by any other document (90 cases, all strata; 22 of 96 against 0 of 64 in the hypoxic stratum); in the unsealed arm, where that document leads only 13 runs, 9 of 162 windows containing it and 43 of 557 without it split (52 of 719 overall), so the correlate is the document leading the window, not its presence. No unsealed hypoxic run uses its case's sealed window (0 of 32), so the hypoxic reversal compares different inputs rather than one input's stability under two arms; "leading document" is confounded with case identity inside the sealed arm, so this is a correlate, not a controlled test.

Pinned-window design and results. From `experiments/campaign_b_pinned.py` over `examples/medical_demo_evidence_pinned.json` (30 cases $\times$ 8 repeats $\times$ 3 calls, 720 calls on the same model, 2026-09-05, 0 failures). For each case the window is fixed at the unsealed arm's epoch-1 repeat-0 window, verified byte-for-byte against the 90 recorded repeat-0 prompts, and everything else follows the sealed path. Its window-lead distribution matches the race's (the oxygen document leads 8 of 240 runs, one case), its distinct evidence windows sit at the floor of 30, and its model inputs number 91 against a floor of 90 (one case whose specialist vote moved on an identical prompt). It flips 6 of 210 consecutive pairs, 3 of 120 disjoint pairs and 2 of 30 cases (bounds 0.0556, 0.0633, 0.1953), each at or below the sealed arm, and the hypoxic stratum flips 0 of 28 against the sealed arm's 6 of 28 (Table 24). Its modal decision equals the unsealed repeat-0 decision in 30 of 30 cases.

Interpretation. These two fixed-window policies both reduce observed disagreement relative to the original unsealed recording. They do not establish equivalence, content-independence or a causal explanation of the hypoxic reversal. The pinned recording was collected later, not interleaved with epoch 1, and only one reference window per case was tested. Pinning preserves those particular reference windows; content ranking avoids needing a reference run. Multiple windows per case and interleaved common-prompt configurations would be needed to separate content-policy and temporal effects. Responses to byte-identical prompts disagree in 53 of 1,680 specialist pairs and 80 of 825 critic pairs, within the same 3–10% range as the sealed arm.

Hold the pair-risk target fixed. Let Z_i be case i's mean pair-disagreement indicator. With equal repeat counts, the mean of Z_i equals the pooled pair fraction and estimates population-average pair risk under iid cases and the declared repeat protocol. Consecutive pairs reuse responses; disjoint pairs avoid reuse but retain case clustering. Neither supplies iid Bernoulli trials merely by being declared exchangeable. From `experiments/cluster_robust_bounds.py`, one-sided 95% sealed limits for consecutive/disjoint pairs are: percentile case bootstrap (10,000 resamples), 0.076/0.067 at 30 cases and 0.070/0.069 at 90; Student-t, 0.077/0.067 and 0.069/0.068; Hoeffding, 0.262/0.257 and 0.175/0.173. The unsealed bootstrap limits are 0.186/0.208 and 0.106/0.097. These estimates are not universally above pooled CP: the sealed disjoint bootstrap and t limits of 0.067 are below 0.0747. Bootstrap and t are approximate and may under-cover with sparse events; neither establishes exact finite-sample validity here. Hoeffding is finite-sample for independent bounded case observations, allowing arbitrary dependence within a case, and does not certify pair risk below 0.1. Its zero-event floor at 30 cases is 0.223.

A separate target. PDR is pair risk without score selection, conditional on both labels being usable. With all executions usable, $\mathrm{PDR} = \Pr(Y_1 \neq Y_2)$ and $R_{\mathrm{any},8} = \Pr(\exists r < 8 : Y_r \neq Y_{r+1})$ are two distinct targets. For iid labels within a case these are $1 - \sum_l p_l^2$ and $1 - \sum_l p_l^8$, averaged over the same case population. Thus the case-any row changes the event as well as the sampling unit. Missing repeats require separate reporting rather than being counted as agreements. Ordinary CP for that row still requires iid case-level Bernoulli events, not exchangeability alone. Neither target is risk conditional on one particular case and its already-issued answer. A certificate must name the loss, case population, repeat protocol, selection rule, epoch and confidence assumptions; the nominal pooled crossings reported here are not such certificates.

Table 25: Document-position association with Campaign A's unusable responses (§4.3), over the unsealed arm's 240 observations. All six unusable responses contain a reasoning block with no answer text, and all six occur in windows whose first retrieved document is the same fraud policy. Three nested conditionings compare position with presence. Fisher one-sided p-values are nominal and exploratory: repeats cluster by ticket, and epoch-1 conditioning was selected after observing failures.

conditioning	**with**	**without**	**nominal** p	$p < 0.05$**?**
policy present anywhere in the window	6/153	0/87	0.065	no
policy in first position	6/81	0/159	0.0013	yes
first against second, among windows containing it	6/81	0/72	0.020	nominally yes

The third row compares positions within the unsealed arm and among windows containing the document. This narrows the association but does not isolate position from ticket content or repeat dependence. The sealed arm retrieves that document 0 times in 240 observations; its arm-level contrast is compatible with, but does not identify, the same mechanism.

Prespecified association in epoch 2. Epoch 1 identified the document by inspecting its six failures, making that conditioning post hoc. Epoch 2 retains the document and the same comparison among windows containing it: 5/59 against 0/68, nominal $p = 0.0197$. Prespecification removes the document-selection step while leaving repeat dependence and confounding unresolved. The document leads fewer windows in epoch 2 (59 against 81); all five unusable answers follow it in first position. The sealed arm retrieves it 0 times in 240 observations in both epochs. Both epochs use `anthropic.claude-fable-5`, twenty days apart.

Clustering and interpretation. Epoch 2's five failures occur on a single ticket; epoch 1's six span three tickets. Within the ticket that fails in both epochs, the comparison is 5/8 against 0/5 ($p = 0.0435$) in epoch 2 and 3/5 against 0/8 ($p = 0.0350$) in epoch 1. Restricting to one ticket removes between-ticket mixing, but the p-values remain nominal because independence of repeats is unestablished. Epoch 1's other two tickets do not individually cross the nominal 0.05 threshold. Thus the association recurs under a fixed hypothesis, with its epoch-2 evidence concentrated in one ticket. Identifying a causal mechanism requires control beyond these observational comparisons. Both epochs, all conditionings and the per-ticket breakdown come from `experiments/positional_channel.py`.

Output-cap comparison. To examine whether reasoning exhausts the 256-token cap, two recordings on 2026-09-05 retain per-response `stop_reason`, output and thinking token counts, and content-block types, which the first two recordings lack. They use the same model, tickets, seeds and design, with 480 calls each at caps of 1,024 and 256 tokens. Every response ends with `end_turn`. Unsealed output-token counts are 6/17/137 (min/median/max) at 1,024 and 6/17/150 at 256, with none at or within 16 tokens of either cap. Each recording has 0 unusable answers of 480. The fixed suspect document leads 71 and 62 unsealed windows, respectively, with no failures; windows containing it outside first position also have 0 failures of 65 and 0 of 57. Sealing produces 12 distinct prompts for 12 tickets against 112 and 111 unsealed. Both arms' flip bounds are 0.0095 (0 flips over 240, one-sided 90%) in both recordings.

The later 256-token run also has no unusable answers, so raising the cap is not necessary for the observed disappearance. This does not uniquely identify a model change: finite-sample variation, different realized windows and other endpoint changes re-

main possible. All four epochs are in `experiments/positional_channel.py` and `experiments/epoch_replication.py`.

Campaign A under two targets and confidence levels (§4.3). Zero sealed flips give a nominal pooled CP limit of 0.0095 over 240 reused within-ticket observations, versus 0.1746 for the distinct ticket-any event over 12 tickets. The latter needs iid ticket events for binomial validity; neither bounds risk conditional on an already-issued answer. Both are 90% bounds, the level the recorder computes (`flip_rate_upper_bound_90pct`, $\delta = 0.10$), unlike Table 4's 95%; at 95% the same counts give 0.0124 and 0.2209. The producer emits all four as `flip_rate_upper_bounds` and checks reproduction of the recorder's original limit before computing the other three. The confidence level and event definition must therefore accompany each quoted limit.

G.1 Epoch replication

To test temporal replication, both §4.3 campaigns were recorded with the same design, seeds and models on 2026-08-26, using 1,920 further billed calls. The comparison retains the original epochs of 2026-08-06 and 2026-08-07; recorders refuse to overwrite committed recordings.

For Campaign A, we compare the arm-specific frequency of unusable answers and the number of distinct prompts. For Campaign B, we compare numerical threshold crossings while retaining each risk target and its interval assumptions. Table 26 reports both recurring and changed observations.

Table 26: Temporal replication of both §4.3 campaigns, approximately twenty days apart, using 1,920 further billed calls. Both epochs are retained for comparison. Campaign A's epochs 3 and 4 were recorded on one day at 1,024- and 256-token caps. Its Fisher tests are nominal observation-level comparisons because repeats cluster by ticket.

	epoch 1	**epoch 2**	**A, epochs 3–4**
A unusable answers, unsealed vs sealed	6 / 240 vs 0 / 240	5 / 240 vs 0 / 240	0 / 240 vs 0 / 240, both
A nominal Fisher one-sided p	0.015	0.031	1.0, both
A distinct prompts, unsealed / sealed	103 / 12	104 / 12	112 and 111 / 12
A tickets returning two different usable labels	0	0	0, both
B sealed bound: consecutive / disjoint / case-level	0.0677 / 0.0747 / 0.2386	0.0677 / 0.0747 / 0.1953	—
B sealed strata clearing ε, by unit	2 / 1 / 0 of 3	2 / 1 / 0 of 3	—
B case-any CP can certify within each stratum	no	no	—

Replicated and changing observations. Campaign A retains 12 sealed prompts in all four epochs by construction. Unusable responses occur only in its unsealed arm in epochs 1–2 and in neither arm in epochs 3–4. Campaign B repeats the sealed counts 8/210 and 4/120 and the 2/1/0 stratum threshold-crossing pattern. Its any-of-eight bound falls from $0.2386 \to 0.1953$ (3 of 30 events, then 2); unsealed bounds also fall. These observations reproduce the numerical threshold pattern, not an invariant arm effect or a certificate.

Flip definition and metric provenance. Flip calculations exclude unusable answers from both numerator and denominator, as Table 2's pN comparison excludes refused pairs. The recorder's historical `n_flipped` field counted `<no-answer>` as a label distinct from the modal answer. Under the usable-label definition, 0 tickets return two different usable labels in both epochs. Reanalysis recomputes both definitions from the same recorded per-repeat decisions without billed calls and verifies the original values first. Both arms then have 0 flips in both epochs; the unsealed nominal flip bound changes from 0.0435 to 0.0098 in epoch 1 and from 0.0383 to 0.0098 in epoch 2, using 234 and 235 usable observations. These are observation-level bounds at 90%, as specified in Appendix G.

Availability denominator. The sealed arm has zero unusable answers, leaving its denominator at 240 and its one-sided 90% bound at 0.0095 under either metric definition. Unusable-response rates retain all billed calls in their denominators, because each call is an opportunity for an availability failure. Thus the epoch-1 contrast remains 6/240 vs 0/240 with nominal $p = 0.015$. Separate denominators preserve the distinction between usable-label agreement and availability.

G.2 THE 90-CASE CAMPAIGN

Larger sample, same risk definitions. To examine the effect of more cases on the disagreement bounds, we extend Campaign B to 90 cases $\times$ 8 repeats $\times$ 2 arms on `anthropic.claude-opus-5`, yielding 4,317 recorded calls. One hypoxic case's unsealed repeat 3 was lost to a read timeout; its billed calls were discarded. That case has 7 observed repeats, leaving 629 adjacent pairs unsealed versus 630 sealed. Table 27 reports the two risk targets separately; the incomplete repeat set is not a fully observed any-of-eight event. Results are produced by `experiments/case_level_certificate.py`.

Table 27: Campaign B at 30 and 90 cases, two different risk targets. Pair-level CP pools adjacent pairs and is nominal; case-any CP asks whether eight repeats disagree and requires iid case events. One unsealed 90-case record has only seven repeats, so its event also has a shorter horizon. All limits are one-sided 95%; numerical threshold crossings are not evidence that arm risks are equal.

campaign	arm	pair-level	upper	case-level	upper	CP limit below 0.1
$n = 30$	unsealed	25/210	0.1623	9/30	0.4651	no, either unit
$n = 30$	sealed	8/210	0.0677	3/30	0.2386	pair-level only
$n = 90$	unsealed	46/629	0.0925	14/90	0.2324	pair-level only
$n = 90$	sealed	29/630	0.0622	12/90	0.2071	pair-level only

Unequal stratum sizes. The 90 cases comprise 51 / 18 / 21 afebrile / febrile / hypoxic cases, meaning cases without fever, with fever, or with low oxygen saturation, respectively. Only one stratum reaches the zero-event minimum $n \geq 29$ needed for a case-any upper limit below the target. Preserving the observed hypoxic proportion would require roughly 125 total cases to yield 29 hypoxic cases in expectation; the realized count could differ.

Sample size with observed flips. The afebrile stratum has 51 cases, of which 4 have a flip: its observed case-any rate is $4/51 = 0.0784$ and its CP upper limit is 0.1706. Per-stratum sealed limits are 0.1706 / 0.3767 / 0.4370, all above 0.1. More independent cases can in principle establish risk below 0.1 even with nonzero flips, if the population risk is sufficiently low. The present observations do not meet that criterion.

Change in the arm contrast. The sealed pair-level rate changes from 0.0381 to 0.0460 and the unsealed rate from 0.1190 to 0.0731, reducing the observed arm contrast at $n = 90$ relative to the 30-case campaign. The unsealed pooled bound is 0.0925, so both nominal pair limits fall below $\varepsilon = 0.1$. The observed rates still differ, and the pooled bounds do not account for case dependence. Changes in case mix, realized races and recording time prevent assigning the smaller contrast to any one mechanism.

The output propagation recurs at scale. From `experiments/cascade_strata.py` on the 90-case recording: the sealed arm's distinct model inputs are 279 against a floor of 270, so 9 of 90 cases send two critic prompts, and 11 of the 12 flipping sealed cases are cases whose specialist votes vary on a byte-identical prompt (the twelfth flips on the critic's own vote). Flipping cases per stratum are 4 of 51 afebrile, 3 of 18 febrile and 5 of 21 hypoxic, so the hypoxic stratum again flips most often; across the 720 sealed runs, 42 depart from their case's modal specialist-vote pattern, 18 change the leader and 20 change the decision.

External validity. Extending these findings requires independent cases and workload diversity as well as repeated executions. The loop uses six constructed cases and a simulated race; CAMELYON isolates image-processing order with no language model and a detector with poor lesion recovery; Campaign B uses synthetic labels with low panel agreement; and Campaign A measures availability and prompt normalization with no usable-label disagreement contrast. The local model control adds evidence from an open-weights model while leaving cross-provider generality unresolved. Together, the studies demonstrate SEAM's recording, intervention and risk-analysis workflow on these mechanisms. Deployment-specific auditing guarantees and clinical claims require representative data and justified sampling assumptions.

H GLOSSARY

Definitions apply throughout the manuscript. The ledger workload's terminal label `refuse` is a completed decision declining settlement, so it counts as a usable answer. An audit refusal means that the audit procedure withholds its requested conclusion.

Table 28: Terms used in this paper.

term	meaning
factor	one named dimension of the configuration a run depends on; client-side or provider-side
sealed arm	an arm that records interactions and normalizes declared client-side orders; provider calls remain live during fresh execution
declared order	the client-side normalization alone: merged fan-in, declared admission, ordered commits, with no recording
audit layer	gateway, content-addressed store, hash-chained trace and comparator; what the sealed arm adds to the declared order
comparator	aligns traces by step, names the earliest differing step, and returns the union of declared reads from that step onward
trajectory τ	the ordered sequence of model calls, tool calls and results in one run
tool sequence	the loop's measured ordered `search_ledger(query,attempt)` strings, including retries but excluding model calls and all results; replay checks equality of this sequence
L_{tool}	tool-attempt count, including retries and excluding model calls
offline replay	re-execution supplied only with recorded occurrences, refusing missing or exhausted requests; no fresh provider sample
execution, run	one end-to-end pass of a workload on one input
re-run, repeat	a further execution of the same case; repeat indexes it within a campaign
bitwise agreement	equality of every bit in the specified output representation
decision agreement	equality across fresh executions of completed answer labels that the workload can interpret
attribution	an intervention-supported factor-set explanation of an observed divergence, or refusal when the procedure withholds an explanation
flip	disagreement between two usable answer labels; missing answers are availability failures
sampling unit	the observation modeled by an interval; repeated pairs cluster within cases. Exchangeability alone does not justify binomial CP inference, even for case-level indicators
selective pair risk $R_g(\lambda)$	population pair disagreement conditional on stratum g, selection by the run-A-only score and both labels being usable; not a bound for each issued answer
any-of-eight risk	population probability of any disagreement across eight repeats of a case; distinct from pair risk
certificate	a statistical guarantee under explicit sampling and theorem assumptions
nominal CP limit	a Clopper–Pearson binomial calculation whose iid Bernoulli assumptions have not been justified for the observations; crossing a target alone cannot certify it
empirical gate acceptance	a candidate passes measured control and single-member-deletion comparisons; validity of the attribution theorem's assumptions requires separate justification
stratum	a pre-declared subgroup of cases within which a bound is computed separately
coverage	probability that an audit procedure returns an answer; for the score rule, the probability that a case's run-A score passes the chosen cutoff, with usable-pair availability reported separately; in attribution, the probability of returning a nonempty factor set

Table 28 (continued)

term	meaning
case	one input of a campaign (a ticket, a clinical presentation, an open amount)
campaign	one recorded design over a set of cases, repeats and arms on one model
arm	one configuration of a campaign: unsealed, sealed, pinned, ordered only
epoch	one recording of a campaign, identified by model, endpoint and date
draw	one recording of the loop campaign; reported ranges describe observed draw-to-draw variation, not population bounds
episode	one run of the loop on one case and repeat
decision	a workload's terminal label; the loop labels are settle, escalate and refuse, while the panel returns a diagnostic label
predicate	one role or check whose output the decision aggregates; a firing rate or label distribution is measured per predicate
verifier	a deterministic classifier or rule replacing a predicate; the term includes label-producing rules as well as proposition checkers, and stochastic upstream inputs can change its output
verifier set	the available replacements; full mechanization purchases all of them, leaving predicates without available replacements unchanged
V_{spec}, V_{critic}, V_{sig}	retrieval-ranked specialist rule, panel-dependent critic rule, and signature-based rule used for any replaced role; no claim of clinical quality
PDR (pair-disagreement risk)	$\Pr(Y_A \neq Y_B \mid U)$, where U means both labels are usable; no score selection. With all executions usable and conditionally iid labels, $\mathbb{E}_X[1 - \sum_l p_l(X)^2]$
pass^k	probability that all k independent attempts succeed at the task, a correctness-based reliability target distinct from agreement
output propagation	an early stochastic output changing downstream inputs, possible in fixed graphs; a loop additionally changes control flow or execution length
tier	one of the attribution benchmark's five difficulty levels, D1–D5
instance	one attribution problem drawn from a synthetic factored oracle with an injected cause (§4.4)
membership test	one subset query to the thresholded divergence oracle, evaluated by fixed repeats or an SPRT; strict mode also counts terminal rechecks
prescribed-action agreement	fraction following the benchmark's D1–D4 answer / D5 refuse policy; not cause-identification accuracy
1-minimal	sufficient under the thresholded oracle, with every single-member deletion insufficient; not necessarily smallest among all sufficient sets
evidence class	[A] analytic results and model calculations; [M] local experimental measurements; [R] billed-endpoint observations, including reanalysis of recorded evidence