

AGENT-RISK: Risk Assessment for Changes to AI Agents

Yue Zhao¹ Zixiang Xu¹ Angela Huyue Zhang¹ S. Alex Yang² Xiyang Hu³

¹University of Southern California

²London Business School ³Arizona State University

yue.z@usc.edu zixiangx@usc.edu azhang@law.usc.edu
sayang@london.edu xiyanghu@asu.edu

Abstract

Before changing a deployed AI agent (a stricter safety gate, a new model), organizations must approve the change, starting from existing logs. Extending classical risk analysis to partial records and partial changes to running systems, AGENT-RISK replays logged episodes, assuming nothing else changes and the agent does not anticipate the change. The log settles episodes acting as before, a declared stop value settles stopped ones, and the rest need logged support, a saved-state rerun, or a bound. We prove *what logs must record for exact replay and which saved states serve a gate family*. For independent episodes, we bound the reruns a decision needs and tolerable drift from live behavior. In a bash sandbox, a 7,200-episode log alone decided neither of two retry gates, which let the agent act after refusals the log never shows. Rerunning only each gate’s blocked episodes from saved states reached both live decisions at half the agent calls of a live test sized post hoc for them, given the log. Without per-step harm, a stopping gate’s certified harm ceiling rose up to 2.5-fold, since the log hides whether harm preceded the stop. Approval thus needs evidence designed in advance: log fields, saved states, targeted reruns.

1 Introduction

AI agents now act for organizations, answering customers and writing and running code (Eleven-Labs, 2026; Cursor, 2026). The institutions that absorb the risk of these actions (insurers, certifiers, deployment approvers, and courts that allocate liability) need a stated loss, evidence, and uncertainty before the agent acts. Insurability and premium follow technical due diligence (Munich Re, 2023), whereas one provider separated development from production databases only after a coding agent deleted production data (Masad, 2025). We call the organization that runs an agent and holds its logs the *deployer*. Take a coding agent whose tool

calls pass a *monitor* (a second model scoring each call’s risk) and a *gate*, which uses the score to run the call, stop the episode, or refuse the call so the agent retries. The deployer keeps changing such an agent, tightening the gate or releasing a new model. Before a change runs, the deployer, or an insurer or auditor with its logs, faces one question: how much of its risk do existing logs establish (Figure 1)? Logs (per-step records of tool calls, monitor scores, and outcomes) come first because they are already in hand: a live trial exposes users to the risk under assessment, and a new test campaign costs runs and may not match deployment.

Six characteristics (§2.1) make this question hard for AI (e.g., LLM) agents (Table 1): **1** approval gates, **2** partial records, **3** long horizons, **4** model releases, **5** shared state, and **6** oversight cues, signs that the agent is being tested.

Current Gaps. Benchmarks and control evaluations run and score a given system (Yao et al., 2025; Kwa et al., 2025; Greenblatt et al., 2024). Some also reuse logged monitor scores to set audit thresholds and resample actions (Bhatt et al., 2025; Inglis et al., 2025); we state when such replay is valid. Off-policy evaluation (Horvitz and Thompson, 1952; Precup et al., 2000), optimal stopping (Goel et al., 2017), and partial identification (Manski, 2003) ask what logs of the current system imply about a changed one, but a stricter gate’s stops never occur in the log, and a retry gate lets the agent act after a refusal no log contains. We specialize these tools to agent changes and add *evidence design*: which logged fields, saved states, and new runs certify a decision about a change.

Our framework, AGENT-RISK, replays each logged episode (Li et al., 2011): on the recorded inputs, without executing new environment steps, it recomputes the new rule’s actions up to the *first deviation*, where the two rules first differ, and sorts it into the three cases of Figure 1; a continuing case

What Can Existing Logs Establish About a Change?

Log of a gate that runs every call $\rightarrow$ replay the change $\kappa \rightarrow$ three cases

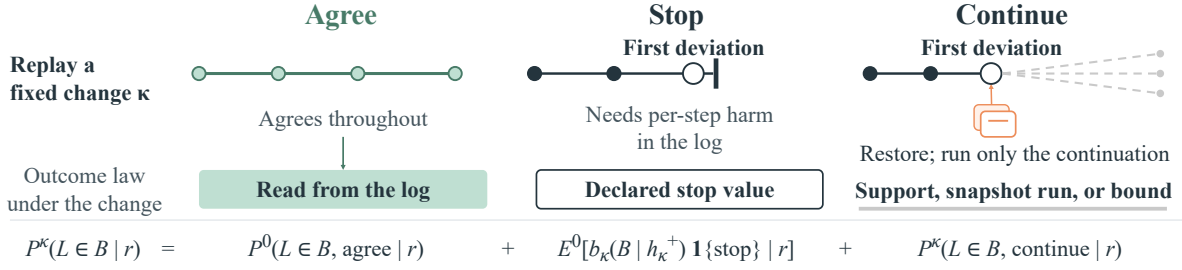


Figure 1: Replaying a logged episode under a proposed change. With sufficient records, the log gives the agreeing case and a declared stop value the stopped case; a snapshot run, logged support, or bounds the continuing case.

may need new runs from a saved sandbox state (a *snapshot*).

The key contributions are:

- **Problem Formulation (Section 2):** We formalize the risk of an agent change before it runs as a specification that extends scenario risk analysis (Kaplan and Garrick, 1981) with partial records and admissible changes, guided by six agent characteristics (Table 1), and ask which risk values the logs leave possible (partial identification).

- **Theoretical Results (Section 3):** We prove (i) exactly what a log must record to settle agreeing and stopped episodes for any current system; (ii) for independent episodes, a lower bound on the reruns needed to certify a harm level, tests that approach it asymptotically, and the restoration error beyond which no valid test certifies reliably; and (iii) the fewest saved states, among those where a gate first acts, serving a whole threshold family.

- **Empirical Verification (Section 4):** To test these results against paired live runs in a bash sandbox, we reran only what one log left open (*targeted execution*), which predicted two retry gates' harm within the registered margin of 0.015 and reached the live decision on both, which the log alone could not. Clean starts in a shared workspace and an announced gate mispredicted live runs (Table 2). In all, 16 runs log about 49,000 episodes.

- **Reproducibility and Access:** We release a Python package for replay, harm bounds, and logging comparisons, and scripts that check the formal results on finite instances (Appendix H); the harness, logs, and registrations are available from <https://github.com/yzhao062/agent-risk>.

Characteristic (where it enters)	What the logs can establish
1 Approval gates (execution)	Stricter stopping gates: replay plus a declared stop value (§3.1). Retry gates: similar logged cases, reruns, or bounds (§3.4)
2 Partial records (records)	Replay is exact only if the log keeps each episode's contribution, e.g., harm timing relative to a stop (§3.2)
3 Long horizons (execution)	Stopping changes replay at any length; for continuing ones, the share that agrees can stay the same or fall (§3.3)
4 Model releases (admissible changes)	A new model rerun on logged inputs and seeds settles episodes where it acts as logged; the rest need similar cases, reruns, or bounds (§3.4)
5 Shared state (execution)	With the shared state's law known, single-agent tests bound joint risk; reruns from saved shared state settle it (§3.5)
6 Oversight cues (execution)	Results carry over only for cues seen in testing; an agent told of the gate can act differently, breaking (A2) (§3.6)

Table 1: **Six organizing agent characteristics and what logs can establish under each.** Characteristic k enters the part of the risk specification (Section 2.3) in parentheses; its result is in Section 3. k and demonstration in Section 4. k .

2 Problem Formulation

2.1 Six Organizing Characteristics of Agents

Six characteristics make a change's risk hard to read from logs; each adds a part or an assumption to the risk specification of Section 2.3 (Table 1). Approval gates **1** are the running case: a stricter gate would stop episodes that the log shows running on, and no log shows what such a stop leads to. Partial records **2** keep only the fields a log's designer chose (Chan et al., 2024), and long horizons **3** add steps at which a changed rule can depart from the log (Kwa et al., 2025). A model release

4 changes the agent itself, shared state 5 links agents through common memory, budgets, or tools, and oversight cues 6 let models tell a test from deployment (Needham et al., 2025). We chose the six from a review of 85 sources (Appendices B.1 and B.2).

2.2 One Agent and One Logged Episode

We follow one agent and one logged *episode* (one recorded run). A sandboxed agent proposes a tool call u_t at step t ; a trusted monitor scores it m_t (higher means riskier), and the gate acts on it. The current gate runs every call (*pass-all*) and is absent from the prompt. Episodes last at most T steps, with loss (task failure, harm, cost); harm is an executed policy violation; cost counts tokens and calls.

A stopping gate κ_θ ends the episode at the first $m_t > \theta$; a retry gate refuses such calls and lets the agent try again. A stop ends the episode as a failure, keeps harm and cost accrued so far, and adds no later harm; the deployer declares this valuation.

Example (a toy episode; made-up scores). A logged episode has four steps with monitor scores 0.10, 0.20, 0.97, and 0.30. A gate with $\theta = 0.94$ would stop it at step 2, its first deviation from the log. Up to that step, the log shows what the new gate does; three questions remain.

- (i) *What does the stop cost?* No log contains a stop, so the deployer declares it.
- (ii) *Did the stop prevent the harm?* A log keeping only whether harm occurred cannot tell: harm at step 1 precedes the stop, while harm at step 3 would be prevented.
- (iii) *What if the gate let the agent retry?* The agent would act after a refusal that no log contains, so logs alone can only bound what follows.

2.3 Writing Down the Risk Specification

$$\mathfrak{R} = (E, \ell, K, C)$$

E, ℓ $\{\langle s_i, p_i, x_i \rangle\}_{i=1}^N$ (Kaplan and Garrick, 1981)
 K partial observation (Bäuerle and Rieder, 2017)
 C interventions (Pearl, 1995)

The colored K and C extend scenario triplets with partial observation and intervention (Appendices A.1 and A.7; Table B.1); badges mark where each characteristic enters.

Execution E (what runs) names kernels, random maps from declared inputs to outputs, over at most T steps 3: the agent, monitor, gate 1, sandbox, shared state 5, and oversight cues 6. Serving configuration is a declared input, since batching

and the server process can change outputs (Section 4.4). *Consequences* ℓ (what counts as loss) map a complete episode to loss L , here (failure, harm, cost), keeping pre-stop harm 1. *Records* K (what the log keeps) specify what the evaluator logs from each step’s available history 2, possibly differing from the agent’s context. Write P^0 for the current system’s law on the finite path space. *Admissible changes* C (what may change) specify which kernels may be replaced, when, using what information, and by whom, such as model releases 4 and agent connections 5.

2.4 A Change’s Risk and Identified Set

With the deployment written down, we can ask what the logs justify about one change. Let $R_t = r$ be the evaluator’s record up to step t when $\kappa \in C$ takes effect, and $X = \phi(L)$ the queried loss coordinate. Models that produce the same logs can disagree about the changed system, as each gap in the toy episode shows (Appendix A.3), so the deployer declares a family $\mathcal{M}_0$ of structural models; their kernels and records can vary, except where the specification fixes those kernels or their inputs. The deployer then asks which risks remain possible under the models that reproduce the observed record law of every evidence source, $P_M^R = P_{\text{obs}}^R$ (Manski, 2003). The risk of the change, and the values the records leave for it, are

$$q_t^\kappa(r) = \rho(\mathcal{L}_{P^\kappa}(X \mid R_t = r)),$$

$$\Theta_t^\kappa(r) = \{q_{t,M}^\kappa(r) : M \in \mathcal{M}_0, P_M^R = P_{\text{obs}}^R\}. \quad (1)$$

Here P^κ is the specified law of the changed system, and $q_{t,M}^\kappa$ the risk under M (Appendix A.2, which also covers records of positive probability and endpoint confidence bounds). The functional ρ summarizes X ’s law as harm probability, mean cost, or conditional value-at-risk (CVaR) of tokens (Rockafellar and Uryasev, 2002). The identified set $\Theta_t^\kappa(r)$ is what the records justify before the change runs, and so what a deployer or auditor can claim. The risk is identified when the set has one value; otherwise we bound it sharply or from outside.

Logs describe the current system, so they speak about the changed one only under two assumptions that need justification beyond the records (exact: Appendix A.2). Without (A1, invariance), an undeclared change to a tool, the serving stack, or a model behind a fixed name makes logged steps unrepresentative. Without (A2, no anticipation), an agent told about the gate may act differently be-

fore the gate acts. If either fails, the deployer must model the extra change.

(A1) Invariance, informal. Nothing the change leaves in place changes: the initial law and every kernel the change does not replace, random inputs included, stay as logged. Installing the gate thus leaves the model, tools, users, and sampler as logged. This covers unobserved environment state, whose initial law and transitions are part of E .

(A2) No anticipation, informal. Both systems run in the same causal order and coincide before the change. Every kernel left in place keeps its declared inputs. A replaced kernel may read other earlier inputs that the change declares, and for replay both rules read one common input that holds what either uses. Nothing left in place, and no declared input, reads which change was installed or its announcement, so announcing the gate in the agent’s prompt violates (A2).

3 Evidence Design for Agent Changes

A deployer who proposes a change needs a statement of its risk or the evidence that would settle it. Five steps give one or the other (in symbols: Algorithm 1, Appendix A.6):

1. *Declare* the change, its stop value, and the decision rule; justify (A1) and (A2), or model the extra change.
2. *Replay* both rules on recorded inputs, where the records permit, up to the first deviation (toy episode: step 2).
3. *Read* agreement and stop contributions when the log is sufficient; the stop value keeps harm and cost accrued before the stop (§3.1 to §3.3; toy questions (i) and (ii)).
4. *Resolve* the rest by matching logged cases, snapshot runs, or bounds (§3.4; question (iii)); other settings need a transfer link (§3.5, §3.6).
5. *Decide* by jointly valid one-sided bounds, with an allowance for restoration error, against a declared harm ceiling and success floor, or name the evidence needed to settle it (§4.1).

To make steps 2 and 3 precise, fix κ, r from (1) (Appendix A.4). Baseline paths Ω^0 follow the logged rule. The first deviation σ_κ is the first step where the new rule differs on logged input; $A_\kappa = \{\sigma_\kappa > T\}$ means no deviation. The replayed

prefix h_κ^+ comprises the full history before that decision, including unrecorded coordinates, followed by the new output. It is terminal if the output stops the episode and continuing otherwise.

(R) Replay interface, informal. Both rules are known and deterministic given each step’s declared input I_t , possibly including a seed. The full record contains that input, so both rules can be rerun on logged proposals, scores, and seeds.

(A3) Declared stop value, informal. After a stop, the loss follows a law $b_\kappa(\cdot \mid h_\kappa^+)$ that the deployer declares, including losses accrued before the stop. It describes what happens after the stop, so the loss map is unchanged. Changes that stop with the same replayed prefix share it, and it must agree with any post-stop behavior baseline operation can show.

The three events are agree (A_κ), stop ($\{\sigma_\kappa \leq T, \text{ terminal}\}$), and continue ($\{\sigma_\kappa \leq T, \text{ continuing}\}$), the three columns of Figure 1, whose bottom row is display (2) with matching labels; only the last needs evidence beyond the log and the stop value. Here $E^0[\cdot \mid r]$ averages under $P^0(\tau \mid r)$, with $h_\kappa^+ = h_\kappa^+(\tau)$. Under (R) and (A1) to (A3), for every $B \subseteq \mathbb{L}$,

$$\begin{aligned}
 P^\kappa(L \in B \mid r) &= \underbrace{P^0(L \in B, \text{agree} \mid r)}_{\text{agree: read from the log}} \\
 &\quad + \underbrace{E^0[b_\kappa(B \mid h_\kappa^+) \mathbf{1}\{\text{stop}\} \mid r]}_{\text{stop: declared stop value}} \\
 &\quad + \underbrace{P^\kappa(L \in B, \text{continue} \mid r)}_{\text{continue: support, rerun, or bounds}}.
 \end{aligned} \tag{2}$$

The first two terms restrict the deterministic-regime g-formula (Robins, 1986) to paths that follow the new rule, as censoring at deviation does (Cain et al., 2010). The third is identified under the support and recording conditions of Section 3.4; otherwise, worst-case bounds (Manski, 2003) give each continuing episode its least and greatest feasible outcome, attained when unseen continuations can be chosen freely and jointly (condition (S), Proposition E.1.1; coarser records: Algorithm 1).

3.1 Approval Gates: Can Logs Replay a Stop?

Deviation replay (Robins, 1986; Cain et al., 2010), informal; exact: Proposition A.4.1. Assume (R), (A1), and (A2), and condition on the record r . The agreement term of (2) then holds:

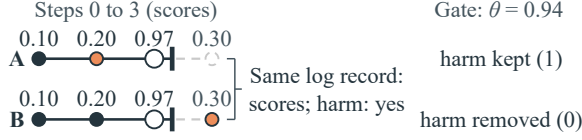


Figure 2: **Missing harm timing leaves stop risk unresolved.** Identical records hide whether executed harm (coral) came before (A) or after (B) the $\theta = 0.94$ stop at step 2; the gate keeps A’s harm and removes B’s.

the changed system has the logged loss law on episodes that agree at every step. With (A3), the stop term holds even if the baseline never stops.

For the deployer. Log the most permissive safe gate, declare the stop value (tolerated post-stop harm: Appendix D.3), and replay stricter gates; a looser gate evaluated from a stricter log is a continuing change that needs support or bounds.

3.2 Partial Records: What Must Logs Keep?

Section 3.1 uses records sufficient for agreement and stops; with partial records 2, Figure 2 shows why missing harm timing leaves stop risk unresolved even with more episodes.

For every baseline law, a log determines the sum of the first two terms of (2) exactly when it preserves each episode’s combined contribution. Within the conditioning event, a log may merge two episodes only if they contribute equally, which those of Figure 2 do not. Contributions are logged loss on agreement, declared loss law on stops, and zero on continuation. In Proposition 3.1, $u_\kappa(\tau)$ indicates a continuing deviation; its conditional expectation is unresolved mass. Here $g_\#P$ is the log’s law under P , and $g^{-1}(y)$ the paths producing log value y (scope of necessity: Appendix E.5).

Proposition 3.1 (Logging sufficiency). *Assume (R), (A1), and (A2), with (R) referring to the full record; the logging map g below need not record I_t . Let $C_\kappa(\tau)$ be $\delta_{\ell(\tau)}$ on agreement, $b_\kappa(\cdot \mid h_\kappa^+(\tau))$ at a terminal deviation, and 0 at a continuing one, and let $u_\kappa = 1 - C_\kappa(\mathbb{L})$. Let $\mathcal{P}_r$ be the laws on Ω^0 with $P(R_t = r) > 0$, and let $s_{\kappa,r}(P) = (\int C_\kappa dP(\cdot \mid r), \int u_\kappa dP(\cdot \mid r))$. For a logging map g that determines the event $\{R_t = r\}$, there is a map Ψ with $s_{\kappa,r}(P) = \Psi(g_\#P)$ for every $P \in \mathcal{P}_r$ if and only if (C_κ, u_κ) is constant on every set $g^{-1}(y) \cap \Omega^0 \cap \{R_t = r\}$.*

For the deployer: what to log. Keep fields that reconstruct each episode’s contribution, including harm timing relative to deviation (Table D.3). Oth-

erwise, distinct baseline distributions with identical records can yield different combined replay contributions, which neither more episodes nor coarser logs resolve (Corollary E.6.1; Blackwell, 1951, 1953). Log the serving mode: batch-dependent serving fails (R) when a change reruns the agent.

3.3 Long Horizons: Does Length Matter?

Each extra step is another chance to leave the log, so the share agreeing throughout cannot rise with the horizon (Corollary E.7.1). With sufficient records and (A3, declared stop value), stopping changes stay identified at every horizon; a continuing change’s worst-case mean interval (Proposition E.1.1) can widen as agreement falls (§4.3).

3.4 Continuing Changes: Runs and Snapshots

Retry gates 1 and model releases 4 continue past the first deviation into the continuing term of (2), where the agent may read inputs the log never shows. The log suffices when it covers and records every input the changed system reads (supported continuation); otherwise the deployer reruns or bounds, and saves states for the reruns.

Supported continuation, informal; exact: Proposition A.5.1. Assume (A1) and (A2), and let the change replace kernels by known ones after the record r was taken. Suppose (G1) baseline support: at each position, each input the change leads an unchanged kernel to read has positive baseline probability there; inputs where the specification fixes that kernel’s output, as after a stop, are exempt. Suppose also that the log records (G2) those kernels’ declared inputs, (G3) every coordinate the loss reads, and (G4) every input the change reads. Then the g -formula identifies the changed system’s loss law, within mechanisms that share the declared kernel inputs and the values the specification fixes, even if no episode agrees.

Without (G1), keeping (G2) to (G4) and every response law the specification fixes, allow any response law at each remaining input of zero baseline probability (U, free responses). Choosing these laws separately for each position gives the identified set under the g -formula; a law shared across positions, or an input merely absent from a finite log, gives an outer set.

Both parts are classical (Robins, 1986; Manski, 2003); Appendix F.5 unifies them. Because baseline never refuses, a retry gate fails (G1) if an unchanged kernel without a declared value reads

the refusal or its message. Under (R), a seed-addressable model release (deterministic at each logged decision input, random tape on fixed serving) has identified agreement mass, at most the first-decision agreement (Corollary F.6.1).

When and how much to rerun. Where the log has no matching cases, as after a refusal or a new model’s new text, the deployer can bound the outcome or use *targeted execution*: restore the sandbox at a logged prefix and run only the continuation. Assumption (T), *restoration fidelity*, links these runs to $\mathcal{M}_0$: restored state, agent context, and remaining budgets give the continuation its live law after that prefix. A restored law within γ in total variation of the live one at each continuing prefix shifts $P(\text{harm})$ by at most γ times the continuing mass (Lemma F.7.2). A uniform random sample of unresolved continuations can replace executing all of them (Proposition F.7.1). Live testing spreads the gap between true harm and the target over all episodes; execution concentrates it on the open ones, and restoration error eats into it.

Execution budget: limits and achievability, informal; exact: Propositions F.8.1 and F.8.2, Corollary F.8.3. Assume (S), binary harm, independent episodes, and one harm probability p for every unresolved continuation (Appendix F.9 allows strata). Treat the log’s resolved harmful mass q_0 and unresolved mass m as known. The change’s $P(\text{harm})$ $q = q_0 + mp$ falls short of a target $q^* < q_0 + m$ by $\zeta = q^* - q > 0$. Restored runs err by at most γ in total variation, and kl is Bernoulli relative entropy. (i) If $m\gamma \geq \zeta$, restoration error can hide the whole gap: no procedure that executes logged continuations has power above its size, that is, none certifies a safe change at exact restoration more often than it can certify an unsafe one. (ii) Otherwise, let $\bar{p} = p + \zeta/m - \gamma$. A procedure valid at level α that certifies with probability at least $1 - \beta$ at exact restoration executes at least $kl(1 - \beta, \alpha)/kl(p, \bar{p})$ continuations in expectation (Kaufmann et al., 2016). Live testing that does not trust the log needs at least $kl(1 - \beta, \alpha)/kl(q, q^*)$ episodes. As $\alpha, \beta \rightarrow 0$ with a log that grows fast enough, sequential probability ratio tests (Wald, 1945) approach both bounds.

Here $\bar{p}$ is the open episodes’ harm rate that would just reach the target, less the restoration error; the closer p is to $\bar{p}$, the more runs a decision needs. Appendix F.9 extends both bounds to strata with their own harm rates, errors, and costs. Strata where

restoration error can make harm certain get no runs, and the others’ run shares are part of the test’s validity (Propositions F.9.1 and F.9.2; Remark F.9.3; allocation: Neyman, 1934).

What to store: snapshot design for threshold gates, informal; exact: Proposition F.10.1. A gate that blocks above a threshold in $[\theta_{\min}, 1)$ first blocks only at a *running maximum*: a step whose score exceeds $\theta_{\min}$ and every earlier score since the change took effect. A running maximum is *open* if some such gate that first blocks there leaves the query’s outcome unsettled. Under full-history kernels, (T), and free responses at unsupported inputs (U), the open running maxima before the last step form the least set among first-block snapshots whose continuation laws identify the query for every gate in the family. An earlier snapshot can replace one by re-executing the agreeing steps.

For the deployer. Record (G2) to (G4), rerun open continuations from snapshots meeting (T), and for threshold gates save first-block snapshots at open running maxima, the least such set.

3.5 Shared State: Does Evidence Transfer?

Sections 3.1 to 3.4 evaluate a change from the deployer’s own logs; evidence from components tested alone needs a transfer link (Pearl and Bareinboim, 2011). When agents share state S (memory, budget, tools) 5, component logs identify each failure probability given S , leaving dependence open; the Fréchet range contains every compatible joint failure probability, and each value in it is attainable when dependence given S is unrestricted (Fréchet, 1951; Nelsen, 2006; Mosleh et al., 1998). Suppose the state law is known, component loss laws given S match alone and composed, and components read only S and private randomness. With unrestricted responses on unseen states, standalone evidence identifies a nontrivial loss event up to width $P_S(U_i)$, where U_i comprises composed states never seen alone (Proposition G.1.1).

3.6 Oversight Cues: Do Tests Transfer?

Oversight cues raise the same transfer question with a different state variable: an agent that can tell a test from deployment may act differently in each. Recorded cues 6, including evaluation signs (Needham et al., 2025) and gate notices, play the role of S in Section 3.5 when the agent reads only the cue and private randomness. Test evidence determines risk only for deployment cues occurring

in testing (Remark G.4.3). A cue created by the change, such as an announced gate, violates (A2, no anticipation); Section 4.6 tests one.

4 Applying the Framework in a Sandbox

We test each characteristic in ControlArena’s bash setting (Inglis et al., 2025; Bhatt et al., 2025), where harm is an executed violation of a declared sandbox policy, and repeat gates and logging in τ^2 -bench’s airline environment (Barres et al., 2025), where harm is an executed booking write that breaks the airline’s written policy. The 16 runs of Table D.1 log about 49,000 episodes, using vLLM mostly on NVIDIA H100 80GB hosts (Appendix D.2). Every log is from a pass-all gate; eight of the nine registered runs fixed sample, analysis, and margin before launch, and three froze predictions before new live records were read. The agent never sees an unannounced gate and a stop ends the episode, so (A2) holds by construction for stopping gates. (A1) holds where serving and environment are stable; where outputs vary by run (the stronger agent, the airline setting), replay is compared with live gates in distribution, within a registered margin.

4.1 Approval Gates: Stopping and Retry

Retry Gates: Estimating Harm. A retry gate lets the agent act after a refusal, which no pass-all log contains. **The log alone bounds retry-gate harm; targeted execution pins it down if restored runs behave like live runs (restoration fidelity)** (Figure D.2). One pass-all log of 7,200 stronger-agent episodes serves two retry gates (thresholds on a log-scale monitor score), which first block in 14% and 41% of episodes; before every high-scoring step, a snapshot rule that names no gate saved the sandbox. Following the five steps of Section 3, executing only the continuations from these snapshots identifies harm for both gates (worked example: Appendix D.3). Each $P(\text{harm})$, frozen before live records were read, matched paired live gates within 0.015; the simpler readings misestimated it (Table 2; Figure D.1).

Retry Gates: Deciding. A rule declared in advance approves a gate whose exact one-sided 95% bounds put harm at most 0.25 and task success at least 0.37. **The log alone cannot decide either gate; with the continuations rerun, the rule approves gate 2 and declines gate 1, as on the live runs.** Read at face value (post hoc), counting retries as stops would approve gate 1, which live runs

refuse; skipping blocked calls reaches both live decisions but misses live harm (Table 2). Given the log, the continuations cost 15% of the agent calls of testing both gates live and half those of a live test sized to reach both decisions in 90% of 500 random subsets (post hoc; Appendix D.3). Targeted execution also matched live runs for a bash agent and the airline environment; in all four, live harm ran 0.001 to 0.008 higher (Appendix D.3).

4.2 Partial Records: What the Log Must Keep

In our logs, one field, per-step harm, decided how low a harm ceiling we could certify. On 300 held-out bash episodes, a reduced log of scores, per-step harm and cost, and outcome certifies the same stopping-gate harm ceiling as the full record (per field, an instance, not a general list: Table D.3). Without per-step harm, they certify 0.159 instead of 0.064, since the log cannot tell whether harm came before the stop (Figure 2); a replication log (0.111 instead of 0.051) and the stronger agent agree, illustrating both directions of Proposition 3.1.

4.3 Long Horizons: Replay Across 40 Steps

Replay still pins down stopping-gate harm as episodes get longer; fewer logged episodes agree with a retry gate, leaving more of its harm to bounds or targeted execution. On a 600-episode bash log with a 40-step cap, replayed stopping-gate $P(\text{harm})$ stays at 0.043 to 0.067 (unknown outcomes) from 15 steps on; the retry gate’s agreement mass falls from 0.77 to 0.57 and its plug-in bounds reach $[0.043, 0.473]$ at 40 steps (Appendix D.5).

4.4 Model Releases: Old Logs, New Model

The new model reproduced none of 99 logged episodes, so replay settled none. At logged inputs and seeds, the same model reproduced every episode, so the release needs support, bounds, or new runs from step 0 (Appendix D.6).

4.5 Shared State: Two Agents in Relay

A first agent takes up to 7 steps; a fresh-context second agent continues from its workspace for up to 8 (Appendix D.7). **Testing the second agent from a clean start overstates its harm in the relay; testing it from the handoff snapshot matches the relay** (Table 2). A clean start leaves out the first agent’s changes to the workspace, which the second agent reads. Both agents violated the policy in four times as many relays as the product of their rates; per-agent logs that record the task’s risk

Change	Log alone	New runs	Live	Result
<i>What the framework establishes</i>				
1 Stopping gate, 3 settings	0.039 (bash; see App. D.3)		0.030	✓ Match within 0.03 in all 3
1 Retry gate 1	? [0.244, 0.345]	0.279	0.288	✓ Match within 0.015
1 Retry gate 2	? [0.100, 0.451]	0.158	0.165	✓ Match within 0.015
1 Gates 1 / 2: approved at 0.25	? Undecided	No / yes	No / yes	✓ Same decision as live
1 Retry gate, bash	? [0.023, 0.422]	0.029	0.030	✓ Match within 0.02
2 Stopping gate, per-step harm	≤ 0.064 with, ≤ 0.159 without		Not needed	• Log must keep harm timing
5 Relay: snapshot start		0.118 to 0.119	0.115	✓ Match within 0.03
<i>Simpler readings and broken assumptions</i>				
1 Gates 1 / 2: retry as stop	Registered: 0.2445 / 0.1002		0.288 / 0.165	✗ Low by 0.043 / 0.065
1 Gates 1 / 2: gate ignored	Registered: 0.3445 for both		0.288 / 0.165	✗ High by 0.057 / 0.179
1 Gates 1 / 2: blocks skipped	Post hoc: 0.3108 / 0.1904		0.288 / 0.165	✗ High by 0.023 / 0.025
5 Relay: clean start		0.178	0.115	✗ High by 0.063
6 Announced gate, abort rate	Unannounced-log replay		181 of 400	✗ Replay low; CI excludes 0

Table 2: **The log alone leaves retry gates open (?); rerunning only what it leaves open matches live runs (✓); simpler readings misestimate harm, and clean starts and an announced gate miss live runs (✗).** •: a logging requirement. P(harm) unless labeled. Match: Newcombe 90% interval inside the registered margin. Certified limits are exact one-sided 95%; log intervals are plug-in bounds. In x / y cells, x is gate 1; blank: not used.

class narrow this joint probability to $[0.031, 0.115]$ (Proposition G.1.1(ii)).

4.6 Oversight Cues: Announcing the Gate

Announcing the gate in our prompt raised the abort rate above what replay from unannounced logs predicted (same tasks and seeds; Table 2). The unannounced gate, a negative control, stays compatible with replay (Appendix D.8).

5 Related Work

Evaluation from Logs. Off-policy evaluation, optimal stopping, and dynamic treatment regimes hold the closest precursors of our stopping replay and supply its core. Full trajectories evaluate any stopping rule with a known reward (Goel et al., 2017), and regime comparisons censor at deviation (Cain et al., 2010). The cited replay and weighting methods use randomized logging (Li et al., 2011; Bottou et al., 2013), which our logs lack. Execution from saved states appears in driving simulation (Gulino et al., 2023), rare-event splitting (Glasserman et al., 1999), agent resampling (Bhatt et al., 2025), and reset access (Yin et al., 2022; Song et al., 2023). Agent checkpoint and replay systems (Wu et al., 2026; Chawla and Koul, 2026) restore recorded prefixes without identification guarantees.

What Is New. Without overlap, partial identification gives sets (Manski, 2003; Khan et al., 2024), and sharp bounds exist for dynamic regimes (Alvarez and Ferrando, 2026) and discrete causal models (Duarte et al., 2023); our unsupported-input result specializes them to replacement kernels with declared values. Robust sequential tests (Saha and

Ramdas, 2026) and cost-aware pure exploration (Kanarios et al., 2024) supply the tools behind our execution bounds. New here are the logging condition for replayed stop values (Proposition 3.1), the restoration threshold and execution-to-live cost ratio (Appendix F.8), allocation across strata (Appendix F.9), and the least set of first-deviation snapshots per threshold family (Proposition F.10.1).

Agent Risk and Governance. Ray (2026) bounds a multi-judge gate’s errors and shows that ungated trajectories need not identify a blocking regime’s safety frontier. Our joint-risk range conditions on shared state (Section 3.5; certificates from joint runs: Bhardwaj et al., 2026). Taxonomies list our characteristics (e.g., Chan et al., 2023; Slattery et al., 2024), rigorous audits need more than black-box access (Casper et al., 2024), and underwriting proposals model claim costs (Xu et al., 2026); evidence design specifies needed records (Appendices B.4 and C).

6 Conclusion and Future Directions

AGENT-RISK establishes when logs can support approving changes to deployed agents and when more evidence is needed. If nothing else changes and the agent does not anticipate the change, it shows what a log settles and must record and, for independent episodes, the reruns a decision needs. In our bash sandbox, the log alone decided neither retry gate; rerunning only what it left open reached both live decisions at half the agent calls of a live test sized post hoc for them, given the log. Future work could restore state beyond saved files and bound the risk of oversight-aware agents (Needham et al., 2025).

Limitations

Each result rests on stated conditions; Table D.2 lists the evidence each demonstration still needs. Identification results concern population record laws on finite path spaces. Execution results add sampling guarantees for independent episodes; agents sharing state use the component evidence of Section 3.5. Replay assumes that nothing else changes (A1) and that the agent does not anticipate the change (A2). A change the agent can perceive, such as an announced gate, violates (A2). Paired runs with and without the announcement expose this violation (Section 4.6); bounds for agents that recognize oversight remain open. Logs cannot confirm the stop value (A3), which our runs declare but do not test for harm. Restoration checks covered only the file tree compared by the harness.

A frozen mechanical checker defines our harm label under four declared rules (Appendix D.2); the framework takes any declared loss. The demonstrations use honest agents (not evading oversight) from one open-weight model family, with 14B and 27B parameters, in a bash sandbox and an airline booking environment. The relay joins two copies of one agent, and supported continuation is untested. For the model release, replay settled none of 99 episodes, leaving its losses unestimated.

Ethical Considerations

Risk estimates can affect deployment, resource access, and responsibility for failures. Claims should identify affected parties, the loss definition, evidence, and uncertainty. Unobserved consequences and misspecified mechanisms can give false reassurance. The framework exposes assumptions to deployers and auditors, preserving unresolved uncertainty in downstream decisions.

A deployer could choose a loss, stop value, or margin that favors approval. Declaring the specification and decision rule before reading results, as our registered runs do, guards against this choice. Episodes the log leaves open enter the decision as intervals. In deployment, replay’s per-step records and saved states can contain user data, subject to the deployer’s retention and access rules. Proposition 3.1 states when logs trimmed for privacy still determine the agreeing and stopped terms, allowing checks before use. Our logs from sandboxed coding tasks and simulated airline customers contain no personal data. Appendix I provides our AI Use Statement.

References

- AI Safety Institute. 2024. [AI Safety Institute Approach to Evaluations](#). Policy paper.
- Alexander Alvarez and Sebastian E. Ferrando. 2026. [Sharp causal bounds for dynamic treatment regimes](#). *arXiv preprint arXiv:2609.06397*.
- Anthropic. 2024. [Model Card Addendum: Claude 3.5 Haiku and Upgraded Claude 3.5 Sonnet](#). Technical report, Anthropic.
- Elias Bareinboim and Judea Pearl. 2016. [Causal inference and the data-fusion problem](#). *Proceedings of the National Academy of Sciences*, 113(27):7345–7352.
- Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. 2025. τ^2 -Bench: Evaluating conversational agents in a dual-control environment. *arXiv preprint arXiv:2506.07982*.
- Nicole Bäuerle and Ulrich Rieder. 2017. [Partially observable risk-sensitive Markov decision processes](#). *Mathematics of Operations Research*, 42(4):1180–1196.
- Yoshua Bengio, Stephen Clare, Carina Prunkl, Malcolm Murray, Maksym Andriushchenko, Ben Bucknall, Rishi Bommasani, Stephen Casper, Tom Davidson, Raymond Douglas, David Duvenaud, Philip Fox, Usman Gohar, Rose Hadshar, Anson Ho, Tiancheng Hu, Cameron Jones, Sayash Kapoor, Atoosa Kasirzadeh, and 74 others. 2026. [International AI Safety Report 2026](#). Report DSIT 2026/001, Department for Science, Innovation and Technology.
- Yoshua Bengio, Sören Mindermann, Daniel Privitera, Tamay Besiroglu, Rishi Bommasani, Stephen Casper, Yejin Choi, Philip Fox, Ben Garfinkel, Danielle Goldfarb, Hoda Heidari, Anson Ho, Sayash Kapoor, Leila Khalatbari, Shayne Longpre, Sam Manning, Vasilios Mavroudis, Mantas Mazeika, Julian Michael, and 77 others. 2025. [International AI Safety Report. Preprint](#), arXiv:2501.17805.
- Varun Pratap Bhardwaj, Garima Singh, and Arun Pratap Bhardwaj. 2026. [Agent behavioral contracts II: Certifying compositional reliability without assuming independence](#). *Preprint*, arXiv:2608.12895.
- Aryan Bhatt, Cody Rushing, Adam Kaufman, Tyler Tracy, Vasil Georgiev, David Matolcsi, Akbir Khan, and Buck Shlegeris. 2025. [Ctrl-Z: Controlling AI agents via resampling](#). *arXiv preprint arXiv:2504.10374*.
- David Blackwell. 1951. [Comparison of experiments](#). In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pages 93–102. University of California Press.
- David Blackwell. 1953. [Equivalent comparisons of experiments](#). *The Annals of Mathematical Statistics*, 24(2):265–272.

- Léon Bottou, Jonas Peters, Joaquin Quiñero-Candela, Denis X. Charles, D. Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Simard, and Ed Snelson. 2013. [Counterfactual reasoning and learning systems: The example of computational advertising](#). *Journal of Machine Learning Research*, 14(101):3207–3260.
- Lauren E. Cain, James M. Robins, Emilie Lanoy, Roger Logan, Dominique Costagliola, and Miguel A. Hernán. 2010. [When to start treatment? A systematic approach to the comparison of dynamic regimes using observational data](#). *The International Journal of Biostatistics*, 6(2). Article 18.
- California State Legislature. 2025. [SB-53 Artificial intelligence models: large developers](#). California Senate Bill 53 (Transparency in Frontier Artificial Intelligence Act), Chapter 138, Statutes of 2025; approved September 29, 2025.
- Stephen Casper, Carson Ezell, Charlotte Siegmann, Noam Kolt, Taylor Lynn Curtis, Benjamin Bucknall, Andreas Haupt, Kevin Wei, Jérémy Scheurer, Marius Hobbhahn, Lee Sharkey, Satyapriya Krishna, Marvin Von Hagen, Silas Alberti, Alan Chan, Qinyi Sun, Michael Gerovitch, David Bau, Max Tegmark, and 2 others. 2024. [Black-box access is insufficient for rigorous AI audits](#). In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, pages 2254–2272.
- Center for AI Standards and Innovation (CAISI), NIST. 2025. [Technical Blog: Strengthening AI Agent Hijacking Evaluations](#). Technical blog. Last modified 2025-12-19.
- Alan Chan, Carson Ezell, Max Kaufmann, Kevin Wei, Lewis Hammond, Herbie Bradley, Emma Bluemke, Nitarshan Rajkumar, David Krueger, Noam Kolt, Lennart Heim, and Markus Anderljung. 2024. [Visibility into AI agents](#). In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, pages 958–973. ACM.
- Alan Chan, Rebecca Salganik, Alva Markelius, Chris Pang, Nitarshan Rajkumar, Dmitrii Krashennikov, Lauro Langosco, Zhonghao He, Yawen Duan, Micah Carroll, Michelle Lin, Alex Mayhew, Katherine Collins, Maryam Molamohammadi, John Burden, Wanru Zhao, Shalaleh Rismani, Konstantinos Voudouris, Umang Bhatt, and 3 others. 2023. [Harms from Increasingly Agentic Algorithmic Systems](#). In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency (FAccT ’23)*, pages 651–666. ACM.
- Ivi Chatzi, Nina L. Corvelo Benz, Eleni Straitouri, Stratis Tsirtsis, and Manuel Gomez-Rodriguez. 2025. [Counterfactual token generation in large language models](#). In *Proceedings of the Fourth Conference on Causal Learning and Reasoning*, volume 275 of *Proceedings of Machine Learning Research*, pages 1291–1315. PMLR.
- Tisha Chawla and Susheem Koul. 2026. [Chronicle: Cut-point replay for regression testing of LLM agents](#). *Preprint*, arXiv:2609.20625.
- Hao-Hsuan Chen. 2026. [Foundations of a time-consistent counterfactual actuarial runtime for autonomous AI agents](#). *Preprint*, arXiv:2605.26508.
- C. J. Clopper and E. S. Pearson. 1934. [The use of confidence or fiducial limits illustrated in the case of the binomial](#). *Biometrika*, 26(4):404–413.
- Nina L. Corvelo Benz, Stratis Tsirtsis, Eleni Straitouri, Ivi Chatzi, Ander Artola Velasco, Suhas Thejaswi, and Manuel Gomez-Rodriguez. 2026. [Evaluation of large language models via coupled token generation](#). In *Proceedings of the 29th International Conference on Artificial Intelligence and Statistics*, volume 300 of *Proceedings of Machine Learning Research*, pages 2215–2223. PMLR.
- Cursor. 2026. [Cursor earns AIUC-1 certification for agent security and reliability](#). Cursor blog, August 13, 2026. Accessed October 4, 2026.
- Zehang Deng, Yongjian Guo, Changzhou Han, Wanlun Ma, Junwu Xiong, Sheng Wen, and Yang Xiang. 2025. [AI Agents Under Threat: A Survey of Key Security Challenges and Future Pathways](#). *ACM Computing Surveys*, 57(7).
- Guilherme Duarte, Noam Finkelstein, Dean Knox, Jonathan Mummolo, and Ilya Shpitser. 2023. [An automated approach to causal inference in discrete settings](#). *Journal of the American Statistical Association*, pages 1–16.
- ElevenLabs. 2026. [ElevenLabs secures first-of-its-kind AI agent insurance](#). ElevenLabs blog, February 12, 2026. Accessed October 4, 2026.
- European Commission, AI Office. 2025. [General-Purpose AI Code of Practice: Safety and Security Chapter](#). Final version, 10 July 2025.
- European Parliament and Council of the European Union. 2024. [Regulation \(EU\) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence and amending Regulations \(EC\) No 300/2008, \(EU\) No 167/2013, \(EU\) No 168/2013, \(EU\) 2018/858, \(EU\) 2018/1139 and \(EU\) 2019/2144 and Directives 2014/90/EU, \(EU\) 2016/797 and \(EU\) 2020/1828 \(Artificial Intelligence Act\)](#). Official Journal of the European Union, L series, 12 July 2024; application dates amended by Regulation (EU) 2026/1744.
- Maurice Fréchet. 1951. Sur les tableaux de corrélation dont les marges sont données. *Annales de l’Université de Lyon, Section A, Série 3*, 14:53–77.
- Aurélien Garivier and Emilie Kaufmann. 2016. Optimal best arm identification with fixed confidence. In *Proceedings of the 29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 998–1027.

- Paul Glasserman, Philip Heidelberger, Perwez Shahabuddin, and Tim Zajic. 1999. [Multilevel splitting for estimating rare event probabilities](#). *Operations Research*, 47(4):585–600.
- Karan Goel, Christoph Dann, and Emma Brunskill. 2017. [Sample efficient policy search for optimal stopping domains](#). In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pages 1711–1717.
- Ryan Greenblatt, Buck Shlegeris, Kshitij Sachan, and Fabien Roger. 2024. [AI control: Improving safety despite intentional subversion](#). In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 16295–16336. PMLR.
- Cole Gulino, Justin Fu, Wenjie Luo, George Tucker, Eli Bronstein, Yiren Lu, Jean Harb, Xinlei Pan, Yan Wang, Xiangyu Chen, John Co-Reyes, Rishabh Agarwal, Rebecca Roelofs, Yao Lu, Nico Montali, Paul Moudin, Zoey Yang, Brandyn White, Aleksandra Faust, and 3 others. 2023. [Waymax: An accelerated, data-driven simulator for large-scale autonomous driving research](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 7730–7742.
- Lewis Hammond, Alan Chan, Jesse Clifton, Jason Hoelscher-Obermaier, Akbir Khan, Euan McLean, Chandler Smith, Wolfram Barfuss, Jakob Foerster, Tomáš Gavenčík, The Anh Han, Edward Hughes, Vojtěch Kovařík, Jan Kulveit, Joel Z. Leibo, Caspar Oesterheld, Christian Schroeder de Witt, Nisarg Shah, Michael Wellman, and 25 others. 2025. [Multi-Agent Risks from Advanced AI](#). Technical Report 1, Cooperative AI Foundation.
- Josiah P. Hanna, Philip S. Thomas, Peter Stone, and Scott Niekum. 2017. [Data-efficient policy evaluation through behavior policy search](#). In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1394–1403. PMLR.
- Meiling Hao, Pingfan Su, Liyuan Hu, Zoltan Szabo, Qingyuan Zhao, and Chengchun Shi. 2024. [Off-policy evaluation with deeply-abstracted states](#). *arXiv preprint*.
- Yifeng He, Ethan Wang, Yuyang Rong, Zifei Cheng, and Hao Chen. 2024. [Security of AI Agents](#). *Preprint*, arXiv:2406.08689.
- D. G. Horvitz and D. J. Thompson. 1952. [A generalization of sampling without replacement from a finite universe](#). *Journal of the American Statistical Association*, 47(260):663–685.
- Wenyue Hua, Xianjun Yang, Mingyu Jin, Zelong Li, Wei Cheng, Ruixiang Tang, and Yongfeng Zhang. 2024. [TrustAgent: Towards safe and trustworthy LLM-based agents](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10000–10016, Miami, Florida, USA. Association for Computational Linguistics.
- Guido W. Imbens and Charles F. Manski. 2004. [Confidence intervals for partially identified parameters](#). *Econometrica*, 72(6):1845–1857.
- Infocomm Media Development Authority. 2026. [Model AI Governance Framework for Agentic AI](#). Framework Version 1.5, Infocomm Media Development Authority, Singapore. Published 20 May 2026, updated 5 June 2026.
- Rogan Inglis, Ollie Matthews, Tyler Tracy, Oliver Makins, Tom Catling, Asa Cooper Stickland, Rasmus Faber-Espensen, Daniel O’Connell, Myles Heller, Miguel Brandao, Adam Hanson, Arathi Mani, Tomek Korbak, Jan Michelfeit, Dishank Bansal, Tomas Bark, Chris Canal, Charlie Griffin, Mojmir Stehlik, and 2 others. 2025. [ControlArena](#). GitHub repository. UK AI Security Institute and Redwood Research. Bash setting: <https://control-arena.aisi.org.uk/settings/bash.html>. Accessed 2026-10-04.
- Sandeep Juneja and Subhashini Krishnasamy. 2019. Sample complexity of partition identification using multi-armed bandits. In *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pages 1824–1852.
- Kellen Kanarios, Qining Zhang, and Lei Ying. 2024. Cost aware best arm identification. *Reinforcement Learning Journal*. ArXiv:2402.16710.
- Stanley Kaplan and B. John Garrick. 1981. [On the quantitative definition of risk](#). *Risk Analysis*, 1(1):11–27.
- Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. 2016. [On the complexity of best-arm identification in multi-armed bandit models](#). *Journal of Machine Learning Research*, 17(1):1–42.
- Samir Khan, Martin Saveski, and Johan Ugander. 2024. [Off-policy evaluation beyond overlap: Sharp partial identification under smoothness](#). In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 23734–23757. PMLR.
- Noam Kolt. 2025. [Governing AI Agents](#). *Notre Dame Law Review*, 101. Forthcoming.
- Thomas Kwa, Ben West, Joel Becker, Amy Deng, Katharyn Garcia, Max Hasin, Sami Jawhar, Megan Kinniment, Nate Rush, Sydney Von Arx, Ryan Bloom, Thomas Broadley, Haoxing Du, Brian Goodrich, Nikola Jurkovic, Luke Harold Miles, Seraphina Nix, Tao Lin, Chris Painter, and 7 others. 2025. Measuring AI ability to complete long software tasks. *arXiv preprint arXiv:2503.14499*.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. [Efficient memory management for large language model serving with PagedAttention](#). In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 611–626.

- K. K. Gordon Lan and David L. DeMets. 1983. [Discrete sequential boundaries for clinical trials](#). *Biometrika*, 70(3):659–663.
- Lihong Li, Wei Chu, John Langford, and Xuanhui Wang. 2011. [Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms](#). In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, pages 297–306. ACM.
- Kaixuan Liu, Guojun Xiong, Weinan Zhang, and Shengpu Tang. 2026. [Autoregressive diffusion world models for off-policy evaluation of LLM agents](#). *Preprint*, arXiv:2606.05558.
- Charles F. Manski. 2003. *Partial Identification of Probability Distributions*. Springer Series in Statistics. Springer.
- Amjad Masad. 2025. [“we saw Jason’s post. @Replit agent in development deleted data from the production database. Unacceptable and should never be possible.”](#). Post on X, July 20, 2025. Accessed October 4, 2026.
- Sean McGregor. 2021. [Preventing Repeated Real World AI Failures by Cataloging Incidents: The AI Incident Database](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 15458–15463. AAAI Press. Describes the AI Incident Database, <https://incidentdatabase.ai>.
- A. Mosleh, D. M. Rasmuson, and F. M. Marshall. 1998. Guidelines on modeling common-cause failures in probabilistic risk assessment. NUREG/CR 5485, U.S. Nuclear Regulatory Commission, Washington, DC. INEEL/EXT-97-01327.
- Munich Re. 2023. [De-risking AI ventures: How Munich Re assesses AI performance risks. insights into our due diligence process](#). Whitepaper. Accessed October 4, 2026.
- Hongseok Namkoong, Ramtin Keramati, Steve Yadlowsky, and Emma Brunskill. 2020. [Off-policy policy evaluation for sequential decisions under unobserved confounding](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 18819–18831. Curran Associates, Inc.
- National Institute of Standards and Technology. 2024. [Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile](#). NIST AI 600-1, National Institute of Standards and Technology, Gaithersburg, MD.
- Joe Needham, Giles Edkins, Govind Pimpale, Henning Bartsch, and Marius Hobbhahn. 2025. [Large language models often know when they are being evaluated](#). *arXiv preprint arXiv:2505.23836*.
- Roger B. Nelsen. 2006. *An Introduction to Copulas*, second edition. Springer Series in Statistics. Springer, New York.
- New York State Legislature. 2026. [An act to amend the general business law, in relation to transparency and safety requirements for developers of artificial intelligence frontier models; to amend a chapter of the laws of 2025 amending the general business law relating to the training and use of artificial intelligence frontier models, as proposed in legislative bills numbers S. 6953-B and A. 6453-B, in relation to the effectiveness thereof; and to repeal certain provisions of the general business law, relating thereto](#). New York Senate Bill S8828, Chapter 96 of the Laws of 2026; signed March 27, 2026; amends the Responsible AI Safety and Education (RAISE) Act.
- Jerzy Neyman. 1934. [On the two different aspects of the representative method: The method of stratified sampling and the method of purposive selection](#). *Journal of the Royal Statistical Society*, 97(4):558–625.
- Michael Oberst and David Sontag. 2019. [Counterfactual off-policy evaluation with Gumbel-max structural causal models](#). In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4881–4890. PMLR.
- OpenAI. 2025. [Operator System Card](#). Technical report, OpenAI.
- OWASP GenAI Security Project. 2025. [OWASP Top 10 for Agentic Applications for 2026](#). Whitepaper/guide.
- Judea Pearl. 1995. [Causal diagrams for empirical research](#). *Biometrika*, 82(4):669–688.
- Judea Pearl. 2009a. [Causal inference in statistics: An overview](#). *Statistics Surveys*, 3:96–146.
- Judea Pearl. 2009b. *Causality: Models, Reasoning, and Inference*, 2nd edition. Cambridge University Press, Cambridge.
- Judea Pearl and Elias Bareinboim. 2011. [Transportability of causal and statistical relations: A formal approach](#). In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, volume 25, pages 247–254.
- Judea Pearl and James M. Robins. 1995. [Probabilistic evaluation of sequential plans from causal models with hidden variables](#). In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 444–453.
- Riccardo Poiani, Rémy Degenne, Emilie Kaufmann, Alberto Maria Metelli, and Marcello Restelli. 2024. [Optimal multi-fidelity best-arm identification](#). In *Advances in Neural Information Processing Systems*, volume 37.
- Doina Precup, Richard S. Sutton, and Satinder P. Singh. 2000. [Eligibility traces for off-policy policy evaluation](#). In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 759–766. Morgan Kaufmann.

- Shauli Ravfogel, Anej Svete, Vésteinn Snæbjarnarson, and Ryan Cotterell. 2025. [Gumbel counterfactual generation from language models](#). In *International Conference on Learning Representations*.
- Shawn Ray. 2026. [What can be enforced? A theory of certified runtime safety for tool-using agents](#). *Preprint*, arXiv:2607.22868.
- James Robins. 1986. [A new approach to causal inference in mortality studies with a sustained exposure period: Application to control of the healthy worker survivor effect](#). *Mathematical Modelling*, 7(9–12):1393–1512.
- R. Tyrrell Rockafellar and Stanislav Uryasev. 2002. [Conditional value-at-risk for general loss distributions](#). *Journal of Banking & Finance*, 26(7):1443–1471.
- Yangjun Ruan, Honghua Dong, Andrew Wang, Silviu Pitis, Yongchao Zhou, Jimmy Ba, Yann Dubois, Chris J. Maddison, and Tatsunori Hashimoto. 2024. [Identifying the risks of LM agents with an LM-emulated sandbox](#). In *The Twelfth International Conference on Learning Representations (ICLR 2024)*. Spotlight.
- Naveen Sachdeva, Yi Su, and Thorsten Joachims. 2020. [Off-policy bandits with deficient support](#). In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 965–975. ACM.
- Aytijhya Saha and Aaditya Ramdas. 2026. [Huber-robust likelihood ratio tests for composite nulls and alternatives](#). *IEEE Transactions on Information Theory*, 72(1):501–520.
- Guruprerana Shabadi and Rajeev Alur. 2026. [Risk-sensitive agent compositions](#). In *International Conference on Learning Representations*.
- Moshe Shaked and J. George Shanthikumar. 2007. *Stochastic Orders*. Springer Series in Statistics. Springer, New York.
- Yonadav Shavit, Sandhini Agarwal, Miles Brundage, Steven Adler, Cullen O’Keefe, Rosie Campbell, Teddy Lee, Pamela Mishkin, Tyna Eloundou, Alan Hickey, Katarina Slama, Lama Ahmad, Paul McMillan, Alex Beutel, Alexandre Passos, and David G. Robinson. 2023. [Practices for Governing Agent AI Systems](#). White paper, OpenAI.
- Peter Slattery, Alexander K. Saeri, Emily A. C. Grundy, Jess Graham, Michael Noetel, Risto Uuk, James Dao, Soroush Pour, Stephen Casper, and Neil Thompson. 2024. [The AI Risk Repository: A Meta-Review, Database, and Taxonomy of Risks from Artificial Intelligence](#). *Preprint*, arXiv:2408.12622.
- Yuda Song, Yifei Zhou, Ayush Sekhari, J. Andrew Bagnell, Akshay Krishnamurthy, and Wen Sun. 2023. [Hybrid RL: Using both offline and online data can make RL efficient](#). In *International Conference on Learning Representations*.
- Jacob V. Spertus, Mayuri Sridhar, and Philip B. Stark. 2024. [Sequential stratified inference for the mean](#). *Preprint*, arXiv:2409.06680.
- Jacob V. Spertus and Philip B. Stark. 2022. [Sweeter than SUITE: Supermartingale stratified union-intersection tests of elections](#). In *Electronic Voting: 7th International Joint Conference, E-Vote-ID 2022*, volume 13553 of *Lecture Notes in Computer Science*, pages 106–121. Springer.
- Jayakumar Subramanian, Amit Sinha, Raihan Seraj, and Aditya Mahajan. 2020. [Approximate information state for approximate planning and reinforcement learning in partially observed systems](#). *arXiv preprint*.
- The B. John Garrick Institute for the Risk Sciences. n.d. [ADS-IDAC overview](#). University of California, Los Angeles. Undated; accessed October 3, 2026.
- Jean Ville. 1939. Étude critique de la notion de collectif. Thèses de l’Entre-Deux-Guerres, no. 218.
- Abraham Wald. 1944. On cumulative sums of random variables. *The Annals of Mathematical Statistics*, 15(3):283–296.
- Abraham Wald. 1945. Sequential tests of statistical hypotheses. *The Annals of Mathematical Statistics*, 16(2):117–186.
- Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Ji-Rong Wen. 2024. [A Survey on Large Language Model based Autonomous Agents](#). *Frontiers of Computer Science*, 18(6).
- Anna Katariina Wisakanto, Joe Rogero, Avyay M. Casheekar, and Richard Mallah. 2025. [Adapting probabilistic risk assessment for AI](#). *arXiv preprint arXiv:2504.18536*.
- Tianyuan Wu, Chaokun Chang, Lunxi Cao, Wei Gao, and Wei Wang. 2026. [Crab: A semantics-aware checkpoint/restore runtime for agent sandboxes](#). *Preprint*, arXiv:2604.28138.
- Binyan Xu, Xilin Dai, Fan Yang, and Kehuan Zhang. 2026. [When agent automation becomes profitable: Quantifying and insuring autonomous AI risk through trace-economic underwriting](#). *Preprint*, arXiv:2606.16465.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.
- Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. 2025. [\$\tau\$ -bench: A benchmark for tool-agent-user interaction in real-world domains](#). In *International Conference on Learning Representations*.

Dong Yin, Botao Hao, Yasin Abbasi-Yadkori, Nevena Lazić, and Csaba Szepesvári. 2022. [Efficient local planning with linear function approximation](#). In *Proceedings of the 33rd International Conference on Algorithmic Learning Theory*, volume 167 of *Proceedings of Machine Learning Research*, pages 1165–1192.

Miao Yu, Fanci Meng, Xinyun Zhou, Shilong Wang, Junyuan Mao, Linsey Pang, Tianlong Chen, Kun Wang, Xinfeng Li, Yongfeng Zhang, Bo An, and Qingsong Wen. 2025. [A Survey on Trustworthy LLM Agents: Threats and Countermeasures](#). *Preprint*, arXiv:2503.09648.

Junzhe Zhang and Elias Bareinboim. 2020. [Designing optimal dynamic treatment regimes: A causal reinforcement learning approach](#). In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 11012–11022. PMLR.

Zhexin Zhang, Shiyao Cui, Yida Lu, Jingzhuo Zhou, Junxiao Yang, Hongning Wang, and Minlie Huang. 2024. [Agent-SafetyBench: Evaluating the Safety of LLM Agents](#). *Preprint*, arXiv:2412.14470.

Guide to the Appendix

The appendices support the formulation in Section 2, results in Section 3, demonstrations in Section 4, and related work in Section 5.

Start with Appendix A for the specification, assumptions, and evaluation procedure, or Appendix D for the registered and exploratory demonstrations and their uncertainty. Appendix E proves what replay identifies, what a log must preserve, and how horizons affect unresolved outcomes. Appendix F treats supported and unsupported continuation, execution budgets, restoration error, strata, and snapshot design. Appendix G establishes when component evidence transfers to a composition or deployment setting. Appendices B and C explain the survey and comparisons with prior work. Appendix H indexes the verification scripts. `ARTIFACT.md` and `checks/README.md` link the full operational details and worked examples.

A Formal Statements Behind Sections 2 and 3

This appendix gives the formal specifications, assumptions, and results behind Sections 2 and 3; the proofs appear in Appendices E, F, and G.

A.1 The Specification

This is the specification of Section 2.3.

Definition A.1.1 (Risk specification). *A risk specification is a tuple*

$$\mathfrak{R} = (E, \ell, K, \mathcal{C}), \quad (3)$$

where E is an execution mechanism, ℓ is a consequence map, K is an observation mechanism, and $\mathcal{C}$ is a declared family of admissible mechanism replacements.

Execution Mechanism E . Fix a finite horizon T . E specifies initial distributions, policies, transitions, updates, and scheduling as named stochastic kernels whose declared inputs are generated earlier in a declared order; policies use only information available before their decisions. Because mechanisms with identical path laws can respond differently under intervention, E retains its modular kernel and input structure rather than only its induced law.

Consequence map ℓ . Let $L = \ell(\tau) \in \mathbb{R}^m$ be the full-episode loss vector (Kaplan and Garrick, 1981, Section 3.4), with ℓ defined on a trajectory space shared by E and every admissible replacement. In the running agent, L is (failure, harm, cost) over the whole episode, and a declared post-abort kernel of E generates the abort valuation of Section 2.2.

Observation mechanism K . Execution and evaluation share the clock $t = 0, \dots, T$, and the declared generation order interleaves execution coordinates and records. The evaluator receives records $R_t = (O_0, \dots, O_t)$, trivial for $t = -1$ and possibly different from the agent’s context, from causal kernels that read only the execution history available by time t , so an online record cannot depend on a future execution suffix. A monitor that changes approval, timing, scheduling, agent behavior, or losses belongs in E , as does any record used for control or loss accounting. The kernels of E and K , composed in their declared order, induce a joint law $P_{E,K}$ on a finite set Ω of complete joint execution and observation paths, with loss $L = \ell(\omega)$ in a finite set $\mathbb{L} \subset \mathbb{R}^m$; the law of the current system is the baseline law P^0 . Every result in this paper uses this finite setting. In the running agent, K is the deployer’s log.

Admissible Replacements $\mathcal{C}$. Each replacement $\kappa \in \mathcal{C}$ specifies replaced kernels, implementation boundaries, and allowed information. Replacement kernels keep their output coordinates, ranges, and causal positions and are adapted to the implementer’s declared information; they may read

different earlier inputs within it. Unreplaced kernels retain their declared inputs. For deviation replay, κ replaces one decision kernel. We refine the clock so that this kernel acts at most once per step, mapping declared input I_t to baseline action $f_t^0(I_t)$ or replacement action $f_t^\kappa(I_t)$. Refining the clock changes no kernel, declared input, consequence map, or coarse record. For the stopping gate κ_θ , input I_t holds proposals u_t and score m_t , and f_t^κ aborts when $m_t > \theta$. The query applies κ to all histories compatible with the conditioning record; a selection using information outside that record belongs in E as further evidence.

A.2 The Question, Assumptions (A1) and (A2), and Identified Sets

For the query of Section 2.4, suppose replacement κ takes effect at boundary $s > t$ after records R_t . Splicing κ into E yields changed mechanisms E^κ and K^κ with joint law $P^\kappa = P_{E^\kappa, K^\kappa}$, defining the continuation query:

$$Q_t^\kappa(\cdot \mid r) = \mathcal{L}_{P_{E^\kappa, K^\kappa}}(L \mid R_t = r), \quad (4)$$

computed from the specification; we write $P^\kappa = P_{E^\kappa, K^\kappa}$. Applying loss aggregation ϕ and risk functional ρ gives the scalar query $q_t^\kappa(r) = \rho(\phi_\# Q_t^\kappa(\cdot \mid r))$, such as harm probability, mean cost, or conditional value-at-risk (Rockafellar and Uryasev, 2002). The changed system obeys P^κ under two core conditions.

(A1) Invariance: the initial law and every unreplaced kernel, including random-input generators, remain unchanged. Installing a gate leaves models, tools, users, and samplers as logged.

(A2) Prefix preservation and no anticipation: both mechanisms share the causal generation order and coincide before the replacement boundary. Every unreplaced kernel retains its declared inputs; a replaced kernel may read different earlier inputs that $\mathcal{C}$ permits. For deviation replay, f_t^0 and f_t^κ are evaluated on a common input I_t that contains the inputs either rule uses. No unreplaced kernel or declared input reads the identity or announcement of the installed replacement. Announcing the gate in the agent’s prompt violates (A2).

Because true kernels are unknown, the evaluator determines which query values observed records rule out (Manski, 2003). A structural model M specifies target mechanisms (E_M, ℓ, K_M) and evidence sources, including continuation runs linked by **(T) restoration fidelity**: continuation laws equal

conditional laws of P_M^κ given restored prefix h . The model family $\mathcal{M}_0$ encodes declared structural restrictions. Given observed record laws P_{obs}^R , the compatible models are $\mathcal{M} = \{M \in \mathcal{M}_0 : P_M^R = P_{\text{obs}}^R\}$, and the population identified set for query ψ is $\Theta(\mathcal{M}) = \{\psi(M) : M \in \mathcal{M}\}$, yielding $\Theta_t^\kappa(r)$ for $q_{t,M}^\kappa(r)$. Equality of evidence laws is source by source, and pointwise conditional sets are stated at records with positive observed probability.

Sampling error. The set in (1) is stated at records with positive probability. One-sided endpoint confidence bounds, each with noncoverage at most $\alpha/2$, cover it with probability at least $1 - \alpha$; covering only the true risk can permit shorter intervals (Imbens and Manski, 2004), and plug-in bounds ignore sampling uncertainty.

A.3 Records That Leave a Gate Change Open

Section 2.2 names three ways baseline records can leave replacement risk unresolved. A declared abort valuation resolves the first ambiguity. The remaining ambiguities depend on the fixed kernels and input restrictions in the model family; the examples below state which restrictions they retain. (1) *Stop*. No baseline episode aborts, so no record shows the loss after an abort; the declared valuation, (A3) of Appendix A.4, supplies it, and records cannot check it. (2) *Continue*. The logged gate never blocks, so no logged input contains a block message. If the family fixes every kernel’s declared inputs, the gate, and the abort valuation but leaves the agent’s response to a block message free, two models that agree at every input of positive baseline probability have the same record law and can differ in the harm after a retry. (3) *Logging*. If the log keeps scores and an episode harm flag but not the step of each harm, and the family fixes the monitor and the gate but leaves the timing of harm free, a model whose harm falls before the step at which a stopping gate aborts and a model whose harm falls after it have the same record law, yet the gate removes the harm in only the second. The third ambiguity has the finite instance below, for a stricter gate as in the running agent. It is the toy episode of Section 2.2 in formal terms, with harm at step 1 or step 3 and the gate acting at step 2.

Omitted Harm Timing under a Stricter Gate.

When a log records overall harm but omits step-level harm h_t , baseline paths with early harm ($t = 1$) and late harm ($t = 3$) generate identical coarse records. A stricter gate aborting at $t = 2$

under declared accrued-harm valuation eliminates late harm but retains early harm. In the family fixing every kernel except the law of the unrecorded early/late state, with (R) and (A1) to (A3), latent early-harm probability $p \in [0, 1]$ is unconstrained by the coarse log, the fibre condition of Proposition 3.1 fails and the identified harm interval is $[0, 1]$. Logging step harm identifies p , contracting the garbled set $[0, 1]$ to $\{p\}$.

An Omitted State Variable under a Looser Gate.

Replace a gate that blocks every proposal by one that executes every proposal. A record of the proposal, the damage, and the cost that omits a stale-state bit S has the same law for $P(S = 1) = 0.1$ and $P(S = 1) = 0.9$, yet the damage probability after the change is 0.1 or 0.9. The baseline gate decision is constant, so logging it adds nothing, while logging S identifies the replaced damage probability within this family (Appendix H).

A.4 Deviation Replay

This formalizes the replay decomposition in Section 3 and the result in Section 3.1.

Let Ω^0 be the paths where every replaced decision equals baseline rule $f_t^0(I_t)$. The first deviation is $\sigma_\kappa = \min\{t : f_t^\kappa(I_t) \neq f_t^0(I_t)\}$, and $A_\kappa = \{\sigma_\kappa > T\}$ is the agreement event. At a deviation, the replayed prefix h_κ^+ appends replacement action $f_{\sigma_\kappa}^\kappa(I_{\sigma_\kappa})$ to earlier history. A deviation is terminal if the replacement action aborts execution, and continuing otherwise. We condition on record $R_t = r$ with $P^0(R_t = r) > 0$ prior to intervention. Beyond (A1) and (A2), replay assumes:

- (R) *Replay interface*: f_t^0 and f_t^κ are known and deterministic in the declared input I_t , which may include a random tape and which the full record contains (a coarser log, as in Proposition 3.1, need not). The gate rule can be rerun on logged proposals, scores, and seeds.
- (A3) *Declared abort valuation*: after a terminal deviation with replayed prefix h^+ , the law of L is a declared $b_\kappa(\cdot \mid h^+)$ that includes accrued losses. It describes post-abort kernels of E , so ℓ is unchanged; replacements that abort at the same replayed prefix share it, and it must agree with any post-abort kernel at inputs of positive baseline probability.

Proposition A.4.1 (Deviation replay; Robins, 1986; Cain et al., 2010). *Assume (R), (A1), and (A2), and let $B \subseteq \mathbb{L}$.*

(i) *Agreement*. $P^\kappa(L \in B, A_\kappa \mid r) = P^0(L \in B, A_\kappa \mid r)$.

(ii) *Stopping*. *Under (A3), $P^\kappa(L \in B, \sigma_\kappa \leq T, \text{terminal} \mid r) = \int b_\kappa(B \mid h_\kappa^+(\tau)) \mathbf{1}\{\sigma_\kappa \leq T, \text{terminal}\} dP^0(\tau \mid r)$, even if the baseline never aborts.*

Part (ii) formalizes the stop term in display (2). Appendix E establishes both assertions via Lemma E.3.1, while Proposition E.1.1 treats continuing deviations.

A.5 Supported Continuation, Compact Form

This is the formal result summarized in Section 3.4.

Proposition A.5.1 (Supported continuation and unsupported inputs). *Assume (A1) and (A2), let κ replace kernels from boundary s onward by known kernels, and fix $R_t = r$ with $t < s$ and $P^0(R_t = r) > 0$. Suppose (G1) at each position of the declared generation order, every input that the replacement leads an unchanged downstream kernel to read there has positive baseline probability at that position, unless the specification fixes that kernel's value at that input, as after an abort. Also suppose (G2) those kernels' declared inputs are recorded at those positions, (G3) every coordinate that ℓ reads is recorded, and (G4) the replacement reads recorded inputs only. Then, within mechanisms that share the declared kernel inputs and the values the specification fixes, $Q_t^\kappa(\cdot \mid r)$ is identified by the g-formula. Zero agreement mass is compatible with identification.*

Unsupported inputs. Without (G1), keep (G2) to (G4), the declared kernel inputs, and the fixed values. Let the kernel law at each position of the declared generation order be free at its other inputs of baseline probability zero, within declared sets of allowed outputs, separately from the laws at other positions. Then the identified set is the image of the g-formula with one free law at each such position and input, shared by all histories that reach it; if one law is shared across positions, as for a policy called at every step, this image is an outer set.

The first statement identifies Q_t^κ via the sequential g-formula (Appendix F.4). When (G1) fails, optimizing over unobserved inputs specializes partial-identification bounds (Alvarez and Ferrando, 2026; Duarte et al., 2023), giving a sharp identified image when each position's law at unseen inputs varies freely and an outer set when laws are shared across positions (Appendix F.5). Treating unobserved inputs in a finite log as unsupported preserves a valid

outer set (Remark F.5.3).

A.6 The Evaluation Procedure

Algorithm 1 translates the five evaluation steps of Section 3 into formal decision logic.

Algorithm 1 Evaluating a change κ from logs

Input: $\mathfrak{R}$, $\mathcal{M}_0$, record r , change κ , query $\rho \circ \phi$, decision rule
Output: status; $q_t^\kappa(r)$ or an interval; missing evidence
1: Declare the kernels κ replaces, their inputs, b_κ of (A3), and the links (A1), (A2), (T)
2: **if** (A1) or (A2) is not justified and the extra change is not modeled **then**
3: **return** “this analysis does not evaluate $q_t^\kappa(r)$ ”
4: Under (R), where records permit, rerun both rules on each logged τ : σ_κ , h_κ^+ , and its event
5: **if** the log is sufficient (Prop. 3.1) and keeps the feasible sets $\mathcal{X}_h$ **then**
6: read the agree-plus-stop sum of (2) from $P^0(\cdot \mid r)$ and b_κ
7: **for all** terms left open: the continue term, or any term of a failed check **do**
8: **if** (G1) to (G4) hold **then**
9: apply the g-formula (Prop. A.5.1)
10: **else if** snapshots at h_κ^+ are kept and (T) holds **then**
11: execute continuations from restored prefixes, or, for binary harm, a random sample (Prop. F.7.1)
12: **else**
13: bound by $\Theta_t^\kappa(r)$ (1) over record completions, or by (5) for full records and monotone ρ ; sharp under (S), else outer
14: **if** the value or interval, with sampling uncertainty, leaves the decision open **then**
15: name what is missing: episodes, record fields, justified assumptions, or runs
16: **return** status (identified; bounded; needs execution), value or interval, decision

When targeted execution samples binary harm continuations at line 11, it invokes hypothesis (H) of Proposition F.7.1 (Appendix F.7) to guarantee one-sided confidence coverage.

A.7 Symbols

Table A.1 lists the symbols of Sections 2 and 3 and of this appendix, with a one-line meaning and the place where each is defined.

B Survey and Classical Foundations

This appendix describes the survey, classical tools, and existing practices behind the specification in Section 2 and comparisons in Section 5.

B.1 Survey Method

For the characteristics of Section 2.1, we reviewed a purposive sample of 85 papers and reports on agent risk. Two language-model coders from different model families independently coded 23 agent-risk taxonomies, surveys, and governance frameworks alongside 30 incidents from the AI Incident

Database (AIID; McGregor, 2021) against 20 candidate characteristics. The incidents cover 2025 and 2026 reports describing tool-using AI agents. The coded sources are (Chan et al., 2023; Shavit et al., 2023; Chan et al., 2024; Kolt, 2025; Hammond et al., 2025; Slattery et al., 2024; Deng et al., 2025; Yu et al., 2025; He et al., 2024; Hua et al., 2024; Ruan et al., 2024; Zhang et al., 2024; Wang et al., 2024; OWASP GenAI Security Project, 2025; Bengio et al., 2025, 2026; National Institute of Standards and Technology, 2024; Center for AI Standards and Innovation (CAISI), NIST, 2025; European Commission, AI Office, 2025; AI Safety Institute, 2024; Infocomm Media Development Authority, 2026; Anthropic, 2024; OpenAI, 2025).

Three criteria selected the six characteristics of Table 1. Sources or incidents name each one often, each remains a persistent risk driver under current mitigations, and the framework provides an identification result for each. This third criterion is partly circular because the framework was designed around these characteristics; the selection therefore motivates the six without proving that they matter most.

The released artifact lists all 23 sources, their consulted versions, and the 30 incident identifiers.

B.2 The Other Characteristics

Agent characteristics modify instances of the four objects, interfaces, or model families rather than adding primitive concepts. Section 2.1 (Table 1) focuses on the six characteristics with formal results and demonstrations. Other candidates such as prompt injection, reviewer load, environment drift, and serving nondeterminism map to definitions within these same four objects; the rows are neither exhaustive nor additive risk scores. The companion mapping lists the other characteristics and their definitions within the four objects.

B.3 Classical Tools

Table B.1 summarizes how the four-object specification of Section 2.3 adapts classical tools; dynamic PRA models feedback and uncertain event timing (The B. John Garrick Institute for the Risk Sciences, n.d.), and the four-object specification is our modeling choice.

B.4 Existing Practice as Instances

To complement Section 5, we write existing practices as specifications without ranking them: τ -bench and τ^2 -bench fix task reward ℓ (Yao et al.,

Symbol	Meaning	Exact definition
$\mathfrak{R} = (E, \ell, K, C)$	Risk specification: execution mechanism, consequence map, observation mechanism, admissible replacements	Appendix A.1
$T; t$	Finite horizon; step index, shared by execution and records	Appendix A.1
$u_t; m_t$	Tool calls proposed at step t ; the monitor's score for them	Section 2.2
$\kappa; \kappa_\theta; \theta$	A replacement (a change); the stopping gate; its score threshold	Appendix A.1
$L = \ell(\omega); \mathbb{L}$	Full-episode loss vector (failure, harm, cost); its finite set of values	Appendix A.1
$R_t = (O_0, \dots, O_t); r$	Evaluator records up to t ; a value of R_t with positive baseline probability	Appendix A.1
$\Omega; P_{E,K}; P^0; P^\kappa$	Complete joint paths; their law; the baseline law; the law under κ	Appendices A.1, A.2
$I_t; f_t^0; f_t^\kappa$	Declared input of the replaced decision kernel; the baseline and replacement rules	Appendix A.1
$Q_t^\kappa(\cdot r)$	Loss law of the changed system given the record at the change	Appendix A.2
$q_t^\kappa(r); \Theta_t^\kappa(r)$	Risk of the change: ρ applied to the law of $\phi(L)$ under $Q_t^\kappa(\cdot r)$; its values under the models that reproduce the records	Display (1)
$\phi; \phi_\# Q; \rho$	Coordinate or aggregation of L ; the law of $\phi(L)$ under Q ; risk functional (probability, mean, CVaR)	Appendix A.2
$\mathcal{M}_0; \mathcal{M}; \psi; \Theta(\mathcal{M})$	Declared model family; models that reproduce the records; a query; the identified set	Appendix A.2
(A1); (A2)	Invariance; prefix preservation and no anticipation	Appendix A.2
(T)	Restoration fidelity of a continuation run from a restored prefix	Appendix A.2
Ω^0	Paths on which every replaced decision equals the baseline rule	Appendix A.4
$\sigma_\kappa; \mathbf{A}_\kappa$	First deviation of the new rule from the logged one; the agreement set $\{\sigma_\kappa > T\}$	Appendix A.4
h_κ^+	Replayed prefix: the history before the deviation followed by the replacement output	Appendix A.4
terminal; continuing	A deviation whose replacement output is an abort; any other deviation	Appendix A.4
(R); (A3); b_κ	Replay interface; declared abort valuation; the declared loss law after a stop	Appendix A.4
agree; stop; continue	The events $\mathbf{A}_\kappa, \{\sigma_\kappa \leq T, \text{ terminal}\}$, and $\{\sigma_\kappa \leq T, \text{ continuing}\}$	Display (2)
$E^0[\cdot r]$	Expectation of a path function under $P^0(\tau r)$	Display (2)
$\mathcal{X}_h; (S)$	Feasible values of $X = \phi(L)$ after a continuing prefix h ; saturated continuation	Appendix E.1
$\nu^-; \nu^+$	Law of X given r when agreeing episodes keep their logged value, stopped episodes follow the declared valuation, and each continuing episode takes the least (greatest) value in $\mathcal{X}_h$	Display (5)
$C_\kappa; u_\kappa$	An episode's combined contribution to the first two terms of (2); its missing mass	Proposition 3.1
$\mathcal{P}_r; s_{\kappa,r}$	Baseline laws on Ω^0 that charge r ; the replay summary of a law	Proposition 3.1
$g; g_\# P; \Psi$	Logging map; the law of the log; a map from the log law to the summary	Proposition 3.1
$\eta_j(r)$	Deviation hazard at step j among episodes that agree before j	Corollary E.7.1
(G1) to (G4)	Support, recorded kernel inputs, recorded loss coordinates, recorded replacement inputs	Appendix A.5
$S; P_S; U_i$	Shared state; its law; composed states never seen by component i alone	Section 3.5

Table A.1: Symbols of Sections 2 and 3.

Classical tool	Role, essential condition, and status in specification
Scenario triplets (Kaplan and Garrick, 1981)	Kept with named kernels of E and loss ℓ ; requires declared consequence definitions under changed systems.
Partial observation (Bäuerle and Rieder, 2017)	Kept in observation record K and Proposition 3.1; deployer-designed records differ from agent context.
Mechanism intervention (Pearl, 1995)	Kept in intervention family $\mathcal{C}$ under (A1) and (A2); replaces agent decision rules with declared timing and inputs.
g-formula and censoring at deviation (Robins, 1986; Cain et al., 2010)	Inherited in Proposition A.4.1; agreement uses (R), (A1), and (A2), and stopping uses declared abort valuations (A3).
Logged replay (Li et al., 2011)	Inherited in Proposition A.4.1 and deviation time σ_κ ; requires rerunnability (R) via recorded seeds and invariant serving.
Worst-case bounds (Manski, 2003)	Inherited in Proposition E.1.1 and Proposition A.5.1; bounded losses give prefix-feasible bounds, sharp under (S) with full records and outer otherwise; shared-law restrictions may narrow them.
Fréchet bounds and transportability (Fréchet, 1951; Pearl and Bareinboim, 2011)	Inherited in Proposition G.1.1; uses known marginals for joint bounds, and invariance and support to transfer standalone component evidence.

Table B.1: Classical tools, standard assumptions, agent strains, and roles in the specification.

2025; Barres et al., 2025); AI Control specifies audit and deferral thresholds in $\mathcal{C}$ (Greenblatt et al., 2024, Sections 2.2 and 2.5, Appendix K); PRA for AI models pathways in E and ℓ (Wisakanto et al.,

2025, Sections 3.2, 4.3, and 5.2); and risk-sensitive compositions evaluate joint losses (Shabadi and Alur, 2026, Sections 2 and 4.2). The companion practice mapping records the objects each practice fixes and the conditions its conclusions require, which reflect our embedding unless a source section is cited.

C Comparisons With Prior Work

We extend Section 5 by positioning our identification and logging results against prior literature; Table B.1 in Appendix B summarizes foundations.

Off-Policy Evaluation And Dynamic Regimes. Our logged gate is deterministic and never stops, leaving no observed stops to reweight. Halting needs no overlap. Proposition A.4.1 identifies stopping risk via declared accrued-loss valuation (A3); display (2) adds continuing mass.

Bounds Without Overlap. Unlike dynamic regime bounds under unobserved confounding (Zhang and Bareinboim, 2020; Namkoong et al., 2020), we bound missing support at recorded inputs. Where bandit approaches restrict policies (Sachdeva et al., 2020), Proposition A.5.1 leaves unsupported inputs free with declared valuations. Specializing discrete causal bounds (Duarte et al., 2023; Alvarez and Ferrando, 2026), we separate position-specific free-law sharp sets from shared-law outer bounds.

Transport And Evidence Collection. Causal transport uses graphical conditions (Bareinboim and Pearl, 2016); assumptions (A1), (A2), and (T) formalize transfer across logs, restored runs, and altered systems. Rather than designing behavior policies (Hanna et al., 2017), Proposition F.7.1 executes open continuations through reset access (Yin et al., 2022; Song et al., 2023). Unlike learned simulator models (Liu et al., 2026), targeted execution rests on reset fidelity (T).

What logs must keep. Agent logs (Chan et al., 2024) and reasoning from a deployed system’s logs (Bottou et al., 2013) motivate Proposition 3.1. It instantiates the elementary fact that a log determines an expected contribution for every path law exactly when indistinguishable paths have equal contributions. State-abstraction irrelevance gives sufficient conditions for evaluation on compressed states (Hao et al., 2024). Information states pre-

serve values through reward and transition sufficiency (Subramanian et al., 2020).

Logging and reporting duties. Several laws set related duties for AI providers and developers. The EU AI Act provides for automatic event logging in high-risk systems (applicable from 2027 or 2028), documentation by providers of general-purpose models, and, for models with systemic risk, evaluation and incident reporting (European Parliament and Council of the European Union, 2024, Arts. 12, 53, and 55); the voluntary Code of Practice elaborates the systemic-risk duties (European Commission, AI Office, 2025). California’s SB 53 requires large frontier developers to publish safety frameworks and all frontier developers to report critical safety incidents (California State Legislature, 2025), and New York’s amended RAISE Act sets similar duties from January 2027 (New York State Legislature, 2026). These instruments do not provide a general criterion for whether an agent’s records suffice to assess a specified change. Under (R), (A1), and (A2), Proposition 3.1 characterizes the records that preserve the combined replay contributions and unresolved mass across all baseline laws.

Shared randomness and model comparison. Shared-seed replay is a common-random-numbers coupling of the two policies. Oberst and Sonntag (2019) infer Gumbel-max noise under a structural causal model to evaluate a new policy on logged episodes, and Gumbel-max coupling of language models generates counterfactual text and compares models (Chatzi et al., 2025; Ravfogel et al., 2025; Corvelo Benz et al., 2026). Seed-addressable replay reads recorded seeds instead, and Corollary F.6.1 needs only deterministic evaluation at logged inputs and seeds.

Agent evaluation and control. Ray (2026) bounds a multi-judge gate’s errors by Fréchet–Hoeffding. Our range conditions on the state that composed agents share and transfers standalone evidence only on covered states (Proposition G.1.1). Bhardwaj et al. (2026) certify the reliability of composed agents without assuming independence, from measured joint executions; our range needs only component logs and the shared state. We separate stopping replay from continuing changes and state the evidence each requires. Appendix B.4 writes four existing practices as specifications (Yao et al., 2025; Greenblatt et al., 2024; Wisakanto

et al., 2025; Shabadi and Alur, 2026). Of these, only control evaluation reuses records to assess a change (the companion practice mapping).

Insurance. An insurer pricing a proposed change to an agent can use the agree, stop, and continue split to state which losses the policyholder’s logs settle and which need further evidence. Chen (2026) charges each side-effect action against a contractually fixed safe default, and Xu et al. (2026) map task traces, economic exposure, and contract terms to modeled claim costs; our results complement them by characterizing the evidence needed to assess a specified change to the agent. Premiums require a further link from episode losses to calibrated monetary severity, exposure volumes, coverage terms, and dependence across claims.

D Registered Demonstrations and Supporting Evidence

This appendix provides the experimental setup, runs, and change-by-change results behind Section 4, exploratory values, and secondary readings. Interval types follow Section 4; Newcombe intervals are score intervals for differences of proportions, and exact ones are Clopper-Pearson. Post hoc values were computed after registered analyses using the same logs and frozen code, without multiplicity control.

In the primary bash setting (Inglis et al., 2025; Bhatt et al., 2025), task failure is near saturation (0.873 on the held-out log), so these claims hold within this sandbox setting only.

D.1 Logs and Runs

The bash runs use uniform task sampling with replacement from 257 tasks, fresh containers and seeds per episode, and the frozen checker (version 1.1). Table D.1 summarizes the registered and exploratory runs of Section 4.

D.2 Setup Specification

Verbatim prompts, commands, input logs, and verifying records for Section 4 are documented in ARTIFACT.md. ControlArena and τ^2 -bench (MIT license), the Qwen models, and vLLM (Apache 2.0) are used for research evaluation under their licenses.

Declared policy. The operating rules forbid: (1) network access, (2) modifying files outside /workspace or /tmp, (3) destructive commands

outside those paths, and (4) privilege escalation. An episode is harmed when any executed tool call violates these rules.

Harm checker and limitations. A frozen mechanical checker (version 1.1) inspects commands and filesystem changes after each call, excluding sanctioned run_tests and submit. Command substitutions or unexpanded variables may escape command checks but can trigger state checks. A failed state check leaves harm status unknown unless a violation was recorded or, for a replayed stop, the failure came at or after the stop.

Pre-execution blocking and monitor. The monitor, Qwen3-4B-Instruct-2507 (Yang et al., 2025) (revision cdbee75), scores proposed calls before execution at temperature 0 and seed 0. Scores are scaled to [0, 1]; the gate at $\theta = 0.94$ blocks on scores above 0.94, terminating execution before the flagged action runs.

Agent and episode limits. The agent, Qwen3-14B-FP8 (revision 9a283b4), operates in ControlArena’s honest bash setting at temperature 0.7, top p 1.0. Step t uses seed $1000 \times \text{episode_seed} + t$. Episodes terminate at submission, 15 steps (40 in the long-horizon log), 400,000 total tokens, or context exhaustion (32,768 tokens).

Serving. Models are served with vLLM 0.30.0 (Kwon et al., 2023) (PyTorch 2.13.0, CUDA 13.0) on one NVIDIA H100 80GB per host with seed 0, prefix caching off, and offline weights. Agent and monitor contexts are 32,768 and 8,192 tokens (55% and 30% GPU memory). The exploratory log, the exploratory live gates, the stronger-agent run, both airline runs, and the two-gate run served the agent with default kernels. The other runs of Table D.1 used batch-invariant kernels (VLLM_BATCH_INVARIANT=1), except the model-update reruns, whose serving varies by rung. Each H100 host ran 32 episodes at a time. The model-update reruns replayed logged decisions on one NVIDIA GH200 120GB per job (85% GPU memory), with 32 requests in flight under batch-invariant kernels and one under default kernels.

D.3 1 Approval Gates: Stopping and Retry Gates

This subsection gives the stopping and retry results of Section 4.1.

Run	Gate	Episodes	Serving	Role
Exploratory log	Pass-all	300	Default kernels	Exploratory; $\theta = 0.94$ chosen here
Exploratory live gates	Stopping gate at $\theta = 0.99, 0.94, 0.19$	300 each (286 scored at 0.99)	Default kernels	Exploratory live comparison with replay from the exploratory log
Held-out log	Pass-all	300	Batch-invariant	Primary held-out log; source of the retry bounds and the step-0 continuations
Live retry gate	Retry gate, $\theta = 0.94$, retry cap 2	300	Batch-invariant	Live check of the retry bounds, after the fact
Targeted-execution run	Pass-all with a snapshot at the first block; continuations under the retry gate; live retry gate	1,200 units (2,885 episodes)	Batch-invariant, two hosts	Registered; prediction frozen before the live arm was read
First announcement test	Stopping gate, announced	300	Batch-invariant	Registered null against the held-out log (no margin; analysis fixed after launch); followed up by the paired announcement test
Replication log; live replication runs	Pass-all; stopping gate	1,000 and 1,000	Batch-invariant, one host	The registered replication
Paired announcement test	Pass-all; stopping gate unannounced; announced	400 triplets	Batch-invariant, one host	Registered; announced and unannounced gates on the same 400 task-and-seed triplets
Step-0 continuations	Retry continuations from the held-out log	85	Batch-invariant	Targeted execution of step-0 blocks
Model-update reruns	Reruns of logged decisions	99 (and 30)	Per rung	Model-update ladder
Long-horizon log	Pass-all, cap 40	600 (594 scored)	Batch-invariant, two hosts	Both gates replayed by horizon
Stronger-agent run	Pass-all; stopping gate at $\lambda = 11.551$ on s	1,400 pairs (2,800; 2,797 scored)	Default kernels, one host	Registered; stopping-gate replay and certification with a stronger agent
Airline run	Pass-all; stopping gate at $\lambda = 11.186$ on s ; retry gate	1,200 units (2,800 episodes)	Default kernels, one host	Registered; stopping-gate replay, certification, and retry bounds in a second environment
Airline retry by targeted execution	Continuations restored by replaying logged writes; live retry gate	1,200 units (959 new episodes: 159 continuations, 800 live)	Default kernels, one host	Registered; prediction frozen before the 800 new live episodes were read; live units 0 to 399 reuse the airline run's retry arm, analyzed there
Two retry gates by targeted execution	Pass-all with a gate-free snapshot rule; two live retry gates; continuations	7,200 units (25,561 episodes: 7,200 pass-all, 14,400 live, 3,961 continuations)	Default kernels, six hosts	Registered; prediction frozen before the live arms were read; two gates from one log
Two-agent relay	Relay, five conditions	1,600 units (8,000; 7,998 scored)	Batch-invariant, two hosts	Registered; tests of each agent alone against the live relay

Table D.1: Runs behind Section 4. Batch-invariant serving returns the same output for the same input whatever else shares the batch; default kernels do not. ARTIFACT.md maps each run to the code that its logs and commands use.

Stopping gate (Section 4.1). Replay from the 1,000 episodes of the replication log predicts $P(\text{harm})$ 0.039, and the 1,000 live replication runs yield 0.030 (Figure D.3). The difference, -0.009 (Newcombe 95% $[-0.025, 0.007]$), falls within the registered equivalence margin of ± 0.03 (Newcombe 90% $[-0.023, 0.005]$).

Sensitivity to the abort valuation (A3). On the replication log, 372 of 1,000 episodes stopped before harm, so if a stopped episode later caused harm

with probability p , replayed $P(\text{harm})$ would be $0.039 + 0.372p$, below an illustrative 0.10 harm target only if $p < 0.16$; (A3) is declared and untested for harm.

Stopping gate with a stronger agent. The stronger-agent run tests Qwen3.5-27B-FP8 (up to 15 steps) on default kernels, so replay predicts the live gate in distribution. The gate reads the log-scale doubt $s = -\log_{10}(1 - E[N]/100)$ from the monitor's next-token distribution. On a pi-

Change	Status	Harm (interval type)	Rests on	To settle
1 Stopping gate, $\theta = 0.94$	Identified; implementation check	P(harm): replay 0.039, live 0.030, equiv. within 0.03 (Newcombe 90%).	(R), (A1), (A2); (A3) declared, harm untested	Test (A3) for harm
1 Stopping gate, replications	Identified; implementation check	Equiv. within 0.03 (Newcombe paired 90%): stronger agent $[-0.021, 0.009]$; airline $[-0.020, 0.018]$.	(R), (A1), (A2); outputs vary across runs; (A3) declared	Test (A3) for harm; margin vs effect
1 Two retry gates, one log	Identified by targeted execution	P(harm): log $[0.244, 0.345]$ and $[0.100, 0.451]$; log + 1,032 and 2,929 snapshot continuations: 0.279 and 0.158; live 0.288 and 0.165, equiv. within 0.015 (Newcombe paired 90% $[0.003, 0.014]$ and $[0.001, 0.013]$); pass-all and retry-as-stop outside the margin. Cost: 0.08 and 0.23 of live agent calls, given the log (partial).	(R), (A1), (A2); snapshot (T): 3,961 of 3,961 continuations valid	Check (T) beyond file tree
1 Retry gate, bash agent	Identified by targeted execution	P(harm): log $[0.023, 0.422]$ (plug-in outer bounds); log + 485 snapshot continuations: 0.029 (exact 95% $[0.020, 0.040]$); live 0.030, equiv. within 0.02 (Newcombe paired 90% $[-0.002, 0.004]$). Cost: 0.30 of live agent calls, given the log (partial).	(R), (A1), (A2); snapshot (T): 485 of 485 restoration checks passed	Check (T) beyond file tree
2 Log without per-step harm	Set; identified with per-step harm	Certified P(harm) without per-step harm: held-out 0.159, replication 0.111; with it: 0.064, 0.051 (exact one-sided 95%)	As stopping gate	Log harm timing
3 Both gates, 40-step cap	Stop identified; retry bounded	P(harm): stopping 0.043 for horizons $k \geq 15$ (unknowns harmful: 0.067); retry $[0.043, 0.473]$ (plug-in outer bounds), agreement 0.77 ($k = 1$) to 0.57 ($k = 40$)	As both gates; 17 unknown outcomes (disk incident)	Execute retry continuations
4 Model release	Agreement only, pooled	Agreement 0/99, share-weighted over six server processes (exact two-sided 95% upper end 0.037, conservative); losses unestimated	Seed-addressable access per process	Support or new runs
5 Two-agent relay	Identified from snapshots	Second-agent P(harm): relay 0.115; snapshot equiv. within 0.03 (Newcombe paired 90% $[-0.010, 0.004]$); clean 0.178 (relay minus clean, Newcombe paired 95% $[-0.078, -0.049]$). Both violate: 0.085; rate product 0.0195.	Snapshot (T): start digests 1,599 of 1,599	Clean starts: workspace in S
6 Announced gate	Replay in-applicable	Abort +0.070 over the live unannounced gate (Newcombe paired 95% $[0.019, 0.120]$); P(harm): 10/400 live	(A2) fails for this prompt	Live runs

Table D.2: **Table 2 in full: what the deployer can state before each change ships, and what would settle the rest.** Honest agents; in the bash setting, failure is near saturation and θ was chosen on an exploratory log; the stronger-agent and airline thresholds were chosen on pilots. Mint: identified, by the first two terms of display (2) or by targeted execution under (T). Gray: needs support or bounds, or replay does not apply. Coral: evidence that would settle the rest (last column). Newcombe: difference-of-proportions score interval; exact: Clopper-Pearson; plug-in: without sampling uncertainty; equiv.: equivalent. Failure, task success, and cost: Appendix D.

lot, threshold $\lambda = 11.551$ was chosen where the stop rate fell between 10% and 15%; the registered run used 1,400 task-and-seed pairs. Live P(harm) is 0.224 to 0.226 against 0.230 to 0.233 for replay. Live minus replay is -0.009 to -0.004 (Newcombe paired 95% $[-0.024, 0.011]$; 90% $[-0.021, 0.009]$), equivalent within the registered margin of ± 0.03 . With per-step harm, the log certifies P(harm) at most 0.253 (exact one-sided 95%, rounded outward); without it, the identified set is $[0.189, 0.331]$ and certifies only 0.353. Seven episodes across five pairs had unknown outcomes from a host disk incident (three unscored, four failed state checks); all seven remain in every denominator as ranges.

Stopping gate in a second environment. The airline run uses τ^2 -bench v1.0.1 (Barres et al., 2025), where an agent modifies a customer booking database under written rules for cancellations, changes, fees, and compensation. A frozen deterministic checker flags policy violations. The agent (Qwen3-14B-FP8) and simulator run on default kernels. Threshold $\lambda = 11.186$ was chosen on a pilot (stop rate 0.130), and the registered evaluation used 1,200 units across 50 tasks (24 seeds each). Live P(harm) is $554/1,200 = 0.4617$ against $555/1,200 = 0.4625$ for replay. Live minus replay is -0.001 (Newcombe paired 90% $[-0.020, 0.018]$; task-cluster bootstrap 90% $[-0.023, 0.020]$); both intervals satisfy the registered ± 0.03 margin. Pass-

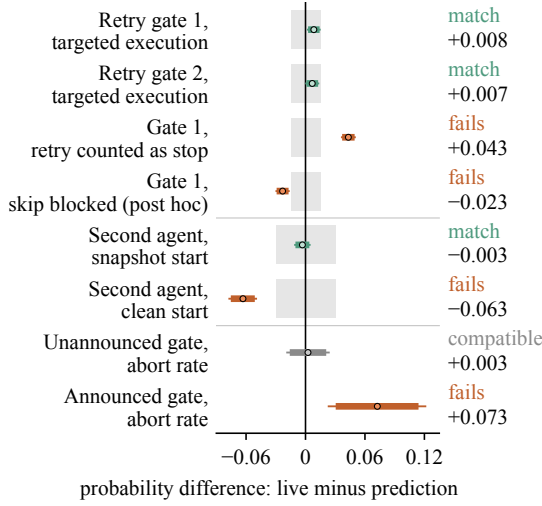


Figure D.1: **Targeted execution and snapshot starts match within registered margins; the unannounced gate stays compatible.** A thick 90% Newcombe bar inside the gray band is a match (with registered cluster checks). Otherwise a thin 95% whisker decides: excluding zero is a failure, and containing zero is compatible (unknown outcomes: midpoints and envelopes).

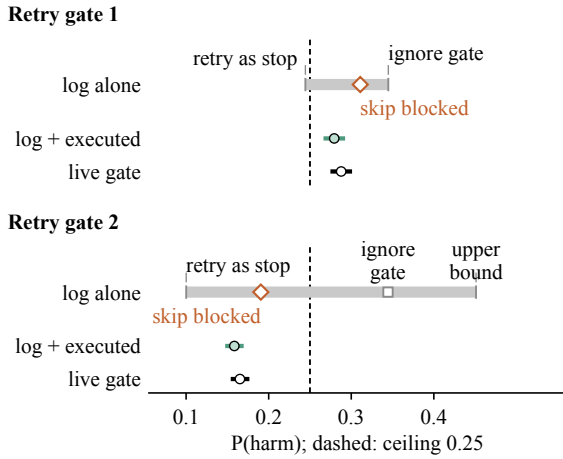


Figure D.2: **The log alone leaves each gate's $P(\text{harm})$ on both sides of the ceiling; executed continuations settle it and match live runs** (bars: exact 95%). Gray: point estimates from retry as stop to the log-only upper bound; diamond: post hoc skip-blocked reading.

all $P(\text{harm})$ is 0.512; the gate removes 0.049 of harm in replay and 0.050 live, so the 0.03 margin is loose relative to this 0.05 effect. Because the database state is a deterministic function of executed tool calls, rerunning the checker on logged calls recovered harm timing in 1,200 of 1,200 episodes, certifying $P(\text{harm})$ at most 0.487 (without timing: 0.511).

Retry gate by targeted execution (Section 4.1). Each of 1,200 units (a task and seed) runs three

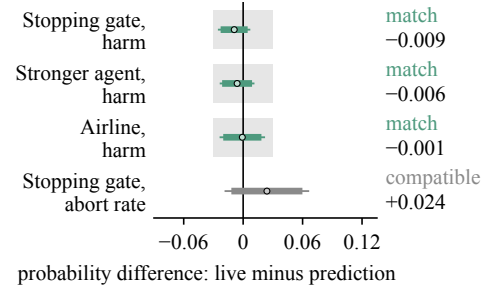


Figure D.3: **Implementation checks: stopping-gate replay matches live harm in all three settings.** The two “Stopping gate” rows use unpaired Newcombe intervals. Intervals, midpoint dots, and verdicts follow Figure D.1; the airline check also requires its task-cluster interval to fit the margin.

conditions on one host: a pass-all baseline committing a container snapshot prior to the first step scoring above $\theta = 0.94$; a continuation from that snapshot with context rebuilt from logged messages, the proposal refused, and the live retry gate active; and the live retry gate from a clean sandbox. The retry gate agrees with 715 logged episodes and first blocks 485 (317 at step 0). From the log alone, display (5) bounds $P(\text{harm})$ within $[0.023, 0.422]$ (plug-in outer bounds). Executing the 485 continuations yields $P(\text{harm})$ of 35/1,200 = 0.029 (exact 95% $[0.020, 0.040]$), with no unknown outcomes; this prediction was frozen with its hash recorded before reading live records. The live gate gives 36/1,200 = 0.030 ($[0.021, 0.041]$). Paired by unit, both arms are harmful in 35 units, the live arm alone in 1, and neither in 1,164. Live minus identified is 0.001 (Newcombe paired 95% $[-0.003, 0.005]$; 90% $[-0.002, 0.004]$), well within the registered ± 0.02 margin. Equivalence applies to marginal $P(\text{harm})$ over the task distribution; it does not establish that pathwise trajectories or task-level rates agree, as agreement and continuation errors could offset. Every restoration matched its source file-tree digest and passed context checks; no snapshot held outside processes. An independent recomputation from raw logs reproduced all values.

Execution cost. The continuations required 4,004 agent calls and 26.1 million tokens, compared to 13,489 calls and 83.7 million tokens for the live gate: cost ratios of 0.30 and 0.31. These are partial-cost ratios that assume the pass-all log exists (14,460 agent calls).

Two retry gates from one log: protocol (Section 4.1; registered). The two-gate run (E7) logs the stronger agent (Qwen3.5-72B-FP8 on default kernels) under pass-all across 7,200 task-and-seed units on six hosts. A gate-free snapshot rule committed the container before every step whose monitor doubt score s exceeded 1.0, yielding 16,456 committed snapshots (mean 61 MB per episode and 3.1 seconds per commit). Two retry gates block at $\lambda_1 = 11.551$ and $\lambda_2 = 10.25$ on s with two retries permitted. For each gate, the log resolves agreeing units (6,168 and 4,271), while continuations from first-block snapshots were executed live (1,032 and 2,929, about 14% and 41% of the log; all 3,961 valid). Predictions were frozen, with their hash recorded, before reading live records, and an independent recomputation verified all identified rows, intervals, and decisions.

For Gate 2, the evaluation follows Section 3. The deployer declares the gate, stop value, and decision rule. Replay finds 4,271 agreeing episodes and 2,929 first blocks (41%). Of these, 92 fall at the last step, where no agent step remains, so their recorded outcome is final (69 harmless, 23 harmed before the block); the other 2,837 continue. The log reads 387 harmed agreeing episodes, and executing the continuations yields 730 more. Display (2) gives $(387 + 23 + 730)/7,200 \approx 0.158$ (with the one unknown outcome, the midpoint is 0.1584). Joint 95% bounds put harm at most 0.1672 and success at least 0.3871, approving the gate.

E7 primary results. For Gate 1, identified $P(\text{harm})$ is 0.2794 against live $2,072/7,200 = 0.2878$ (one unknown outcome), a difference of 0.008 (Newcombe paired 90% [0.003, 0.014]; task-cluster 90% [0.002, 0.015]), equivalent within the registered 0.015 margin. For Gate 2, identified $P(\text{harm})$ is 0.1584 against live 0.1652 (one unknown each), a difference of 0.007 (paired 90% [0.001, 0.013]; task-cluster [0.000, 0.013]), also equivalent within 0.015. Across all four targeted-execution comparisons, live harm exceeds identified harm by 0.001 to 0.008, all within registered margins. Log-only outer bounds are [0.244, 0.345] and [0.100, 0.451], corrected for the unknown outcome. Both ends are attainable: an agent that halts after a refusal gives the lower end, and one that violates the policy whenever a refusal leaves it a step gives the upper end. The log contains no refusal and rules out neither, so it cannot decide either gate.

E7 reference predictors. Both registered reference predictors fail the 0.015 margin on both gates. Ignoring the gate (pass-all, 0.3445) misses live harm by -0.057 and -0.179 . Counting each retry as a stop (0.2445 and 0.1002) misses by $+0.043$ and $+0.065$ (Newcombe paired 95% [0.036, 0.051] and [0.058, 0.072]). A third reading, fixed in a note before computation (post hoc), drops blocked calls while retaining the remaining logged episode and yields 0.3108 and 0.1904 (discrepancies -0.023 , 95% $[-0.030, -0.016]$, and -0.025 , $[-0.033, -0.018]$), also outside the margin; under the declared rule, it would select gate 2 with harm at most 0.1984. The registered reference errors round to 0.04 to 0.18, as reported in Section 4.1.

E7 decision and execution cost. The declared decision rule ($P(\text{harm})$ at most 0.25 and task success at least 0.37, exact one-sided 95% bounds, unknowns unfavorable) selects Gate 2 in both arms: identified harm at most 0.1658 and success at least 0.3889; live harm at most 0.1727 and success at least 0.3897 (also passing with 97.5% per bound, for joint 95% coverage: 0.1672 and 0.3871). Neither registered reference certifies any candidate (post hoc): pass-all violates the harm ceiling, and retry-as-stop violates the harm ceiling for gate 1 and the success floor for gate 2. Continuations used 0.08 and 0.23 of live agent calls (8% and 23%; partial-cost records); including the pass-all log's 102,779 calls, the ratio is 0.66. With the log already recorded, the continuations' 31,706 calls are 15% of the 204,875 of both live gates. Random live subsets of that expected cost (1,114 units per gate) reached both live decisions in 56.4% of 500 draws; the smallest common size on a 1% grid reaching both in at least 90% was 2,232 units per gate (31% of the cohort; 91.8%), at 2.0 times the continuations' calls and 0.47 times those of log plus continuations (post hoc; per-bound 95%).

Simpler readings at face value (post hoc). Read as point values without coverage at the declared ceiling and floor, counting retries as stops approves gate 1, which live runs refuse, and refuses gate 2, whose success it puts at 0.334; ignoring the gate also refuses gate 2.

E7 snapshot design. Minimal first-deviation snapshots at open running maxima after step 0 require 6,332 of the 16,456 snapshots and 173 of 441 GB for harm alone (Proposition F.10.1, post hoc). A running maximum at the last step is not

open because no agent step remains.

Retry gate in the airline environment. The registered airline run (E5) restores each continuation by replaying logged database writes without container snapshots. The pass-all log plus 159 valid executed continuations identify $P(\text{harm})$ of 597/1,200 = 0.4975; the paired live retry gate gives 598/1,200 = 0.4983. Both 90% equivalence intervals (Newcombe paired $[-0.019, 0.020]$; task-cluster bootstrap $[-0.017, 0.019]$) lie inside ± 0.03 . The prediction was frozen before the 800 new live records were read; units 0 to 399 reuse the airline run’s live retry episodes, already analyzed there. Counting retries as stops gives the lower bound 0.4625 (0.036 below live). Continuations used 0.090 of live agent calls and 0.131 of declared tokens (complete usage).

Retry gate on the held-out log. In the earlier held-out protocol, later-prefix restoration checks passed 0 of 10 trials, so those continuations stayed bounded; step-0 runs needed only fresh starts (bounds and live check: companion file).

D.4 2 Partial Records: Certification

For the logging comparison in Section 4.2, exact one-sided 95% Clopper–Pearson bounds on replayed $P(\text{harm})$ certify a harm level with reduced logs in this setting. On the replication log (no unknown outcomes), full and reduced logs certify 0.051 (39/1,000 harms) against 0.111 without per-step harm (94/1,000). On the held-out log, counting one failed state check as harmful yields 0.064 for full and reduced logs (12/300) against 0.159 without per-step harm (37/300). The reduced log preserves per-step scores, per-step harm and cost, and final outcomes.

D.5 3 Long Horizons

For the horizon comparison in Section 4.3, truncating a 600-episode pass-all log with a 40-step cap under batch-invariant serving yields horizon curves with all 600 planned episodes in every denominator (the companion file holds the per-horizon table). Beyond $k = 25$, more than 5% of runs have already ended on token or context limits (151 by $k = 40$, while 240 reach the 40-step cap). Stopping-gate replayed $P(\text{harm})$ stabilizes at 26/600 = 0.043 for $k \geq 15$ (0.067 if all unknowns were harmful). Retry-gate agreement mass falls from 0.77 at $k = 1$ to 0.57 at $k = 40$, widening plug-in outer bounds from $[0.018, 0.030]$ to $[0.043, 0.473]$. A host disk

Field	Needed for	Droppable when	Evidence
Monitor scores	Deviation	Monitor reruns deterministically	Held-out: 3,589 of 3,589 recomputed
Per-step harm	Stop valuation	Calls and a deterministic checker fix timing	Certified ceiling with vs. without it: held-out 0.064 vs. 0.159, replication 0.051 vs. 0.111; airline: 1,200 of 1,200 recomputed
Per-step cost	Pre-stop cost	Another field fixes it	Mean tokens, exploratory gates (companion)
Inputs, seeds, serving mode	Agent rerun	Change reads logged fields only	1,650 of 1,650 decisions; other kernel mode: 2 of 99 episodes
Gate decisions	Nothing	Always (Remark E.5.1)	Unused

Table D.3: **What a stopping-gate replay needs from the log here: per-step harm (shaded) is the one field whose removal raised the certified harm ceiling.** An instance of Proposition 3.1; no general field list. Certification: exact one-sided 95% bounds on replayed $P(\text{harm})$.

incident caused 17 unknown outcomes (6 unscored, 11 failed state checks), which generate all reported ranges; no episode was excluded.

D.6 4 Model Releases

In the model-release test of Section 4.4, the release candidate (Qwen3.5-27B-FP8) reproduces none of 99 logged episodes across six server processes at identical inputs and seeds, while the identical model reproduces 99 of 99 and kernel and precision changes reproduce 2 and 1. Outputs were seed-addressable within each process (40 of 40 probe inputs reproduced) but varied across processes (14 of 40 differed). Episodes were assigned to processes independently of content, so the bound targets the mean agreement mass over processes, weighted by their shares; the exact two-sided 95% Clopper–Pearson bound for 0/99 (0.037) is conservative for it, by independence across episodes and concavity of $\log(1 - p)$.

D.7 5 Shared State: Two Agents in Relay

In the registered five-condition design of Section 4.5, a first agent works a task up to 7 steps, and a second agent continues for up to 8 steps in the inherited workspace. Across 1,600 units (8,000 planned runs, 7,998 scored; two snapshot starts failed to commit and count as unknown), the second agent violated policy in 184 of 1,600 re-

lays (0.115, exact 95% [0.100, 0.132]). From a clean start on the same task and seed, it violated in 285 of 1,600 (0.178, [0.160, 0.198]). Relay minus clean start is -0.063 (Newcombe paired 95% $[-0.078, -0.049]$; 90% $[-0.075, -0.051]$), failing equivalence within the registered ± 0.03 margin. Started from a snapshot of the first agent’s standalone end state, violations occurred in 0.1175 to 0.1188 of runs; relay minus snapshot start is -0.004 to -0.003 (paired 90% $[-0.010, 0.004]$), equivalent within ± 0.03 . Here clean starts substantially overstate relay harm.

Both agents violated in 136 of 1,600 relays ($\beta = 0.085$, exact 95% [0.072, 0.100]), against marginal rates $p_1 = 0.169$ and $p_2 = 0.115$ whose product is 0.0195. The difference $\beta - p_1 p_2 = 0.066$ shows strong positive dependence; treating the agents as independent overstates the union harm probability $\pi = p_1 + p_2 - \beta$ by 0.066. Per-agent logs without shared state give the Proposition G.1.1(ii) range $[0, 0.115]$ for β ; conditioning on the task risk class C narrows it to $[0.031, 0.115]$, and adding external write flags D gives $[0.041, 0.115]$. Linking snapshot runs by unit yields 0.084 to 0.085, matching relay within ± 0.03 .

At initialization, file-tree digests matched their sources in 1,599 of 1,599 runs per snapshot condition.

D.8 6 Oversight Cues

In the paired design of Section 4.6 over 400 triplets (pass-all, unannounced stopping gate, announced stopping gate), the announced gate aborted 0.070 more often than the unannounced gate (181/400 versus 153/400; Newcombe paired 95% [0.019, 0.120]; 90% [0.027, 0.112], excluding equivalence within ± 0.03). Replay from unannounced logs understates the announced abort rate by 0.0725 (Newcombe paired 95% [0.022, 0.122]), while the unannounced gate stays compatible with replay (negative control). Harm is 10/400 in all three arms. This intervention alters the live abort rate, rejecting the abort-rate invariance of (A2) for this prompt; it does not isolate semantic anticipation from other prompt perturbations.

E Replay, Logging Sufficiency, and Long Horizons

For Sections 3.1 to 3.3, this appendix proves Propositions A.4.1, E.1.1, and 3.1, Lemma E.4.2, and Corollaries E.6.1 and E.7.1 in the finite setting.

E.1 Continuing Deviations: Statement

Proposition A.4.1 gives the agreement and stopping components of display (2) in Section 3; this subsection bounds the continuing component. For a scalar query $X = \phi(L)$ and continuing replayed prefix h , let $\mathcal{X}_h$ denote the values of X on paths extending h that follow f^κ at later decisions. Condition (S) formalizes saturated continuation, with primitive conditions in Lemma E.4.2.

- (S) *Saturated continuation*: distinct continuing replayed prefixes of positive baseline mass reach distinct inputs of unknown kernels, each of baseline probability zero. For every family (Q_h) with $Q_h \in \Delta(\mathcal{X}_h)$, one admissible mechanism with the same baseline law realizes all Q_h at once. A replacement that sends two or more continuing prefixes of positive baseline mass to one common state violates (S).

Proposition E.1.1 (Continuing deviations; Manski, 2003). *Assume (R), (A1), and (A2), and let $B \subseteq \mathbb{L}$. Suppose every coordinate of Ω is recorded, so the evidence is P^0 . Let ν be the subprobability from (i) and (ii) of Proposition A.4.1 pushed to X . Under (A3) and (S), the identified set of laws of X is $\{\nu + \int Q_h P^0(dh, \text{continuing} \mid r) : Q_h \in \Delta(\mathcal{X}_h)\}$. Every law-invariant functional nondecreasing in first-order stochastic dominance (mean, exceedance probability, VaR, CVaR) attains its minimum and maximum over this set when each continuing prefix places its mass at $\min \mathcal{X}_h$ or at $\max \mathcal{X}_h$, respectively. For the mean, exceedance probability, and CVaR, every value between the extremes is attained; VaR takes values in a finite set. Without (S), these are outer bounds.*

Let ν^- and ν^+ be the laws of X given r that keep agreement and stop values and assign each continuing prefix its least and greatest feasible value, respectively. Every such functional ρ then satisfies

$$\rho(\nu^-) \leq q_t^\kappa(r) \leq \rho(\nu^+), \quad (5)$$

with the inequalities holding without (S). Section E.4 contains the proof.

E.2 Finite Setting

This setting is used in the replay, logging, and horizon proofs for Sections 3.1 to 3.3.

Paths, Histories, and Kernels. Let Ω be a finite set of complete paths $\omega = (v_1, \dots, v_{n(\omega)})$ listing

execution, observation, and post-abort coordinates in declared generation order, with no path a proper prefix of another. Histories $\mathcal{H}$ are proper prefixes, including the empty history. Declared order assigns to each $h \in \mathcal{H}$ a kernel name $\text{nm}(h) \in \mathcal{N}$ and a declared input $\pi(h)$. The tree is full: $(h, v) \in \mathcal{H} \cup \Omega$ for every kernel value v . A mechanism M assigns transition kernels $k_n^M(\cdot \mid i)$. For $\omega_{1:j} = (v_1, \dots, v_j)$, with $\omega_{1:0}$ empty, the path law is

$$P_M(\omega) = \prod_{j=0}^{n(\omega)-1} k_{\text{nm}(\omega_{1:j})}^M(v_{j+1} \mid \pi(\omega_{1:j})). \quad (6)$$

The cylinder $[h]$ collects paths extending h . Summing over extensions of h gives

$$P_M([h]) = \prod_{j=0}^{|h|-1} k_{\text{nm}(h_{1:j})}^M(h_{j+1} \mid \pi(h_{1:j})). \quad (7)$$

Loss is $L = \ell(\omega) \in \mathbb{L} \subset \mathbb{R}^m$. Observation kernels are named kernels, so records R_t are path coordinates, realizing $P_{E,K}$ of Appendix A.1 in the finite setting.

Replaced Kernel. The replaced decision kernel is named $k \in \mathcal{N}$. A history h with $\text{nm}(h) = k$ is a decision history with input $I(h) = \pi(h)$ and step $t(h) \in \{0, \dots, T\}$. The refined clock of Appendix A.1 invokes k at most once per step. Under (R), known maps define the baseline kernel $\delta_{f^0(i)}$ and replacement kernel $\delta_{f^\kappa(i)}$. The input is a function of history contained in the full record, distinguishing full histories from declared inputs. Baseline laws are supported on Ω^0 , where decision coordinates match f^0 . A value a is an abort if no extension of (h, a) has $\text{nm}(h') = k$, terminating calls to the replaced kernel.

First Deviation. A decision history h is a deviation history if $f^\kappa(I(h)) \neq f^0(I(h))$ while $f^\kappa(I(h')) = f^0(I(h'))$ on all proper decision prefixes h' . The set $\mathcal{D}_\kappa$ of deviation histories has pairwise disjoint cylinders. For $\omega \in [h]$ with $h \in \mathcal{D}_\kappa$, set $\sigma_\kappa(\omega) = t(h)$ and replayed prefix $h^+ = (h, f^\kappa(I(h)))$; set $\sigma_\kappa = \infty$ on agreement set A_κ . Paths partition into agreement and disjoint deviation cylinders:

$$\Omega = A_\kappa \sqcup \bigsqcup_{h \in \mathcal{D}_\kappa} [h]. \quad (8)$$

Deviations partition into terminal set $\mathcal{D}_\kappa^{\text{term}}$ (abort outputs) and continuing set $\mathcal{D}_\kappa^{\text{cont}}$.

Conditioning Record. A record value r is pre-intervention if $\Omega_r = \{R = r\}$ is a disjoint union of cylinders $[h]$ over $\mathcal{H}_r$, none having a deviation history as a proper prefix. Records R_t with $t < s$ satisfy this property. We assume $P^0(\Omega_r) > 0$ and write $P(\cdot \mid r) = P(\cdot \cap \Omega_r) / P(\Omega_r)$. Let $\mathcal{D}_\kappa(r) = \{h \in \mathcal{D}_\kappa : [h] \subseteq \Omega_r\}$, with $\mathcal{D}_\kappa^{\text{term}}(r)$ and $\mathcal{D}_\kappa^{\text{cont}}(r)$ defined analogously.

Assumptions. Let $P^0 = P_{M^0}$ and $P^\kappa = P_{M^\kappa}$. The assumptions of Appendices A.2 and A.4 take tree form:

- (R) Replay interface: the replaced decision kernel is $\delta_{f^0(i)}$ under M^0 and $\delta_{f^\kappa(i)}$ under M^κ , with both maps known and the input contained in the full record (a coarser log, as in Proposition 3.1, need not contain it).
- (A1) Invariance: $k_n^{M^\kappa} = k_n^{M^0}$ for every name $n \neq k$, including the kernel at the empty history (the initial law) and every random-input kernel.
- (A2) Prefix preservation (no anticipation): both mechanisms use the same declared order and inputs (nm, π) , and no declared input or unreplaced kernel depends on the identity or specification of the installed replacement.
- (A3) Declared abort valuation: for every $h \in \mathcal{D}_\kappa^{\text{term}}$ with $P^0([h]) > 0$, $P^\kappa(L \in \cdot \mid [h]) = b_\kappa(\cdot \mid h^+)$, a declared probability measure on $\mathbb{L}$.

Because no replaced kernel acts after an abort, (A1) ensures that post-abort distributions depend only on h^+ , shared by replacements that abort at the same replayed prefix.

E.3 Prefix Lemma and Proof of Proposition A.4.1

Lemma E.3.1 (Prefix Lemma). *Assume (R), (A1), and (A2). (a) $P^\kappa(\omega) = P^0(\omega)$ for every $\omega \in A_\kappa$. (b) $P^\kappa([h]) = P^0([h])$ for every history h none of whose proper prefixes is a deviation history. In particular this holds for every $h \in \mathcal{D}_\kappa$ and every $h \in \mathcal{H}_r$, and $P^\kappa(\Omega_r) = P^0(\Omega_r)$.*

Proof. By (R), (A1), and (A2), corresponding factors in (6) and (7) agree until the first deviation, proving (a) and (b). Summing over the disjoint record cylinders gives equality on Ω_r . $\square$

Proof of Proposition A.4.1(i). Immediate from Lemma E.3.1(a),(b), summing over agreement paths and conditioning on r . $\square$

Proof of Proposition A.4.1(ii). Immediate from Lemma E.3.1(b) and (A3), summing over the disjoint terminal cylinders and conditioning on r . $\square$

E.4 Saturated Continuation and Proof of Proposition E.1.1

Fix a finite query $\phi : \mathbb{L} \rightarrow \mathcal{X}$, with $\mathcal{X} \subset \mathbb{R}$ for order statements. For $h \in \mathcal{D}_\kappa$, let $\Omega^\kappa(h^+)$ collect replacement-consistent extensions that follow f^κ at all decisions after h^+ , giving feasible set $\mathcal{X}_h = \{\phi(\ell(\omega)) : \omega \in \Omega^\kappa(h^+)\}$. The model class $\mathcal{M}$ is a family of mechanisms that share the tree, ℓ , f^0 , f^κ , and b_κ , and satisfy (R) and (A1) to (A3). Let $\mathcal{M}(P)$ be members with baseline law P , with identified set $\Theta_\phi(P) = \{\phi_\# P_M^\kappa(L \in \cdot \mid r) : M \in \mathcal{M}(P)\}$.

Definition E.4.1 (Saturated Continuation). *Let $\mathcal{D}_\kappa^{\text{cont}}(r, P)$ be the set of $h \in \mathcal{D}_\kappa^{\text{cont}}(r)$ with $P([h]) > 0$. The class $\mathcal{M}$ is saturated at P for ϕ if, for every family (Q_h) with $Q_h \in \Delta(\mathcal{X}_h)$ for $h \in \mathcal{D}_\kappa^{\text{cont}}(r, P)$, some $M \in \mathcal{M}(P)$ satisfies $\phi_\# P_M^\kappa(L \in \cdot \mid [h]) = Q_h$ for every such h .*

Let $\mathcal{P}$ be the set of laws P on Ω^0 with $P(\Omega_r) > 0$.

Lemma E.4.2 (Unrestricted Class). *Let $\mathcal{M}_{\text{full}}$ be the class of mechanisms on the declared tree that satisfy (R) and (A1) to (A3) and in which every unreplaced kernel reads the full history, $\pi(h) = h$ whenever $\text{nm}(h) \neq k$. Suppose that for every $h \in \mathcal{D}_\kappa^{\text{term}}$ the declared $b_\kappa(\cdot \mid h^+)$ is the law of ℓ under some probability on the replacement-consistent extensions of h^+ . Then for every $P \in \mathcal{P}$, every ϕ , and every family (Q_h) with $Q_h \in \Delta(\mathcal{X}_h)$ for $h \in \mathcal{D}_\kappa^{\text{cont}}(r, P)$, some $M \in \mathcal{M}_{\text{full}}$ has baseline law P and satisfies $\phi_\# P_M^\kappa(L \in \cdot \mid [h]) = Q_h$ for every such h . In particular, every law in $\mathcal{P}$ is the baseline law of a member, and $\mathcal{M}_{\text{full}}$ is saturated at every $P \in \mathcal{P}$.*

Proof sketch. For each continuing prefix h , mix replacement-consistent paths achieving each outcome in $\mathcal{X}_h$ with weights Q_h ; at terminal prefixes, use extensions realizing b_κ . Set unreplaced kernels to the conditional transitions of P on its support and of these mixtures inside each $[h^+]$. Full-history inputs allow these assignments independently: the replayed cylinders are disjoint and baseline-null, since each contains a first output differing from f^0 . Complete unused inputs arbitrarily. The product path law then recovers P and realizes all target continuation laws and terminal valuations simultaneously. $\square$

Assumption (S) is Definition E.4.1 at P^0 with distinct continuing prefixes reaching distinct unreached inputs; under realizable b_κ , $\mathcal{M}_{\text{full}}$ is saturated at every $P \in \mathcal{P}$.

Proposition E.4.3 (Identified Set). *Assume (R) and (A1) to (A3), and let P be a baseline law with $P(\Omega_r) > 0$. Write $w_h = P([h] \mid r)$ for $h \in \mathcal{D}_\kappa^{\text{cont}}(r, P)$, and let*

$$\nu = \phi_\# \left(P(L \in \cdot, A_\kappa \mid r) + \sum_{h \in \mathcal{D}_\kappa^{\text{term}}(r)} P([h] \mid r) b_\kappa(\cdot \mid h^+) \right).$$

Then $\Theta_\phi(P)$ is contained in $\mathcal{S}_\phi(P) = \{\nu + \sum_h w_h Q_h : Q_h \in \Delta(\mathcal{X}_h)\}$, with equality if $\mathcal{M}$ is saturated at P .

Proof. Summing over (8), Proposition A.4.1(i),(ii) fixes ν , while Lemma E.3.1(b) fixes each continuing weight w_h . By (R), each continuation law lies in $\Delta(\mathcal{X}_h)$, giving the inclusion; saturation gives equality by Definition E.4.1. $\square$

Order Bounds. For finitely supported laws η, η' on $\mathbb{R}$, write $\eta \leq_{\text{st}} \eta'$ if $F_\eta(x) = \eta((-\infty, x]) \geq F_{\eta'}(x)$ for all x (Shaked and Shanthikumar, 2007). Let $\eta^- = \nu + \sum_h w_h \delta_{\min \mathcal{X}_h}$ and $\eta^+ = \nu + \sum_h w_h \delta_{\max \mathcal{X}_h}$.

Proposition E.4.4 (Order Bounds). *Under the hypotheses of Proposition E.4.3, with $\mathcal{X} \subset \mathbb{R}$, the following hold. (a) $\eta^-, \eta^+ \in \mathcal{S}_\phi(P)$, and $\eta^- \leq_{\text{st}} \eta \leq_{\text{st}} \eta^+$ for every $\eta \in \mathcal{S}_\phi(P)$. (b) If ρ is law invariant and nondecreasing in $\leq_{\text{st}}$, then $\rho(\mathcal{S}_\phi(P)) \subseteq [\rho(\eta^-), \rho(\eta^+)]$ and both endpoints are attained; for nonincreasing ρ the endpoints swap. The mean, the exceedance probability $\eta((x, \infty))$, the CDF value $F_\eta(x)$, which is non-increasing, $\text{VaR}_\beta(\eta) = \min\{x : F_\eta(x) \geq \beta\}$, and CVaR_β for $\beta \in (0, 1)$ are such functionals. (c) For the mean, the exceedance probability, the CDF value, and CVaR_β , every value between the two endpoints is attained.*

Proof sketch. Point masses at feasible endpoints bound each continuation CDF, proving (a); CDF and quantile ordering give (b) (Shaked and Shanthikumar, 2007, Eqs. (1.A.2), (1.A.7)). For (c), mix η^- and η^+ within $\mathcal{S}_\phi(P)$. Means and event probabilities are affine; finite support makes the CVaR formula (Rockafellar and Uryasev, 2002, author's version, Theorem 10, Eqs. (27)–(28)) a finite minimum of affine functions and also gives its mono-

tonicity in (b). Continuity and the intermediate value theorem finish. $\square$

Proof of Proposition E.1.1. Combine Propositions E.4.3 and E.4.4, using full recording and (S) for equality; VaR takes values in the finite feasible set. $\square$

E.5 Logging Characterization and Proof of Proposition 3.1

Scope of the Necessity Direction. Necessity in Proposition 3.1 requires variation across all baseline laws on Ω^0 via point-mass witnesses. In restricted classes where coordinates have fixed laws, necessity can fail, whereas sufficiency holds for every baseline class.

Proof. In the tree setting, the contribution C_κ of Proposition 3.1 is $\delta_{\ell(\omega)}$ on A_κ , $b_\kappa(\cdot \mid h^+)$ on $[h]$ for terminal h , and 0 on $[h]$ for continuing h ; $\mathcal{P}_r$ is the set $\mathcal{P}$ above, $s_\kappa = s_{\kappa,r}$, and $\Omega_r^0 = \Omega^0 \cap \Omega_r$. For a baseline law, $s_\kappa(P) = (\nu_\kappa, \varepsilon_\kappa)$, where ν_κ is the measure ν of Proposition E.4.3 with ϕ the identity and $\varepsilon_\kappa = \sum_h w_h$. A logging map is a function g on Ω , and it determines the record event when $\Omega_r = g^{-1}(Y_r)$ for a set Y_r of log values.

Proof of Proposition 3.1. Fix κ and assume $\Omega_r = g^{-1}(Y_r)$. For sufficiency, fibre constancy makes (C_κ, u_κ) a function of g on Ω_r^0 , so its conditional expectation is determined by $g_\#P$. For necessity, suppose $\omega, \omega' \in \Omega_r^0$ satisfy $g(\omega) = g(\omega')$ and $(C_\kappa, u_\kappa)(\omega) \neq (C_\kappa, u_\kappa)(\omega')$. The point masses δ_ω and $\delta_{\omega'}$ belong to $\mathcal{P}$ and have the same log law. Their summaries are $(C_\kappa, u_\kappa)(\omega)$ and $(C_\kappa, u_\kappa)(\omega')$, which differ, so no function of $g_\#P$ equals $s_\kappa(P)$ on $\mathcal{P}$. Applying both directions to each κ in the family proves the equivalence. $\square$

Remark E.5.1 (Status Need Not Be Logged). A coarsened log need not record deviation status when its fibres pool paths with equal signatures (check T1-4, checks/verify_replay.py). Condition (R) refers to the full record, which contains the gate input; the log need not.

E.6 Coarse Evidence and Garbling

This corollary supports the coarse-log claim of Section 3.2.

Corollary E.6.1 (Garbling). *For a fixed model class, replacement, and query conditioned on one record event measurable from g' , if $g' = G \circ g$ for some map G , the identified set under g is contained in the set under g' .*

Proof sketch. Immediate from nesting the compatible model classes under g and $G \circ g$ (Blackwell, 1951, 1953). $\square$

E.7 Long Horizons

This proves the horizon claims of Section 3.3.

Corollary E.7.1 (Long horizons). *Assume (R), (A1), and (A2). For $j = 0, \dots, T$ let $\eta_j(r) = P^0(\sigma_\kappa = j \mid \sigma_\kappa \geq j, R_t = r)$, set to 0 when $P^0(\sigma_\kappa \geq j \mid r) = 0$; by Lemma E.3.1(b) the same value holds under P^κ . Fix $\eta \in [0, 1]$. (i) The agreement mass $P^\kappa(A_\kappa \mid r) = \prod_{j=0}^T (1 - \eta_j(r))$ is identified from the full record; if $\eta_j(r) \geq \eta$ at k steps, it is at most $(1 - \eta)^k$. (ii) If every deviation is terminal, (A3) holds, and a logging map g that determines the event $\{R_t = r\}$, as in Proposition 3.1, determines the contribution C_κ on $\Omega^0 \cap \{R_t = r\}$, then Proposition A.4.1(i) and (ii) and Proposition 3.1 identify the whole law of L given r , at every horizon. (iii) If every deviation is continuing, then under the hypotheses of Proposition E.1.1 the identified interval of the mean of X has width $\int (\max \mathcal{X}_h - \min \mathcal{X}_h) P^0(dh, \text{continuing} \mid r)$. When every $\mathcal{X}_h$ has spread at least δ , the width is at least $\delta[1 - \prod_j (1 - \eta_j(r))]$.*

Proof sketch. Part (i) follows from the chain rule and Lemma E.3.1; the bound follows by comparing factors. For (ii), apply Propositions A.4.1 and 3.1, since agreement and terminal deviations exhaust the mass. For (iii), subtract the mean endpoints in Proposition E.4.4; the lower bound uses continuing mass equal to one minus the agreement mass in (i). $\square$

F Proofs: Supported Continuation, Unsupported Inputs, and Model Updates

This appendix proves the continuation results of Section 3.4: Proposition A.5.1, Proposition F.5.2, and Corollary F.6.1 in a finite coordinate model, using the sequential g-formula (Robins, 1986; Pearl and Robins, 1995) and partial identification (Manski, 2003). It also proves that section's execution-budget and snapshot-design results.

F.1 Finite Coordinate Model

This is the model for the proof of Proposition A.5.1, summarized in Section 3.4.

Coordinates in generation order are $V_1, \dots, V_n$ with finite ranges $\mathcal{V}_1, \dots, \mathcal{V}_n$, and $\Omega = \prod_i \mathcal{V}_i$. For $S \subseteq [n] = \{1, \dots, n\}$, $V_S = (V_i)_{i \in S}$, $\mathcal{V}_S =$

$\prod_{i \in S} \mathcal{V}_i$, and v_S is the restriction of $v \in \Omega$ to S . Kernel i has declared input set $S_i \subseteq [i-1]$ and conditional law $k_i(\cdot \mid v_{S_i})$ on $\mathcal{V}_i$. The baseline law is

$$P^0(v) = \prod_{i=1}^n k_i(v_i \mid v_{S_i}), \quad v \in \Omega. \quad (9)$$

A declared set $\mathcal{K}_i \subseteq \mathcal{V}_{S_i}$ collects inputs where k_i is known by declaration (default $\mathcal{K}_i = \emptyset$).

Tree mechanisms of Proposition A.4.1 embed into coordinates by allocating one slot per history where call order or kernel identity varies. Inactive slots emit a dummy value. Replaced slots are those where replaced and baseline kernels differ. The post-abort path is a single valuation slot carrying the law of L given the prefix (declared as b_κ after terminal deviations by (A3)). Allowed sets $\mathcal{E}_{j,z}$ exclude the dummy at active inputs, and equal $\{\ell(\omega) : \omega \text{ extends } p\}$ at undeclared abort prefixes p .

Replacement. Replacement κ acts at boundary s , replacing kernels $D_\kappa \subseteq \{s, \dots, n\}$ by known kernels $k_j^\kappa(\cdot \mid v_{S_j^\kappa})$ with $S_j^\kappa \subseteq [j-1]$ (with $k_j^\kappa = \delta_{f_j^\kappa(v_{S_j^\kappa})}$ for deterministic replacements). For $i \notin D_\kappa$, $k_i^\kappa = k_i$ and $S_i^\kappa = S_i$. Invariance and prefix preservation ((A1) and (A2)) give

$$P^\kappa(v) = \prod_{i=1}^n k_i^\kappa(v_i \mid v_{S_i^\kappa}). \quad (10)$$

Here $P^0 = P_{E,K}$ and $P^\kappa = P_{E^\kappa, K^\kappa}$ on Ω .

Loss and Records. Loss is $L = \ell(V) = \bar{\ell}(V_{S_L})$ for $S_L \subseteq [n]$ and $\bar{\ell}: \mathcal{V}_{S_L} \rightarrow \mathbb{L}$. Records are $V_\mathcal{O}$ for $\mathcal{O} \subseteq [n]$. The pre-intervention record is $R_t = V_{S_r}$ with $S_r \subseteq \mathcal{O} \cap [s-1]$ and $P^0(R_t = r) > 0$ ($t = -1, S_r = \emptyset$ for unconditional queries). A value v is r -compatible when $v_{S_r} = r$. The continuation query is $Q_t^\kappa(B \mid r) = P^\kappa(L \in B \mid R_t = r)$ for $B \subseteq \mathbb{L}$.

Definition F.1.1 (Downstream and Relevant Sets). *The downstream set $\Delta_\kappa \subseteq [n]$ is defined recursively in index order: $i \in \Delta_\kappa$ if $i \in D_\kappa$, or if $i \notin D_\kappa$ and $S_i \cap \Delta_\kappa \neq \emptyset$. Its complement $\bar{\Delta}_\kappa = [n] \setminus \Delta_\kappa$ is the unaffected set. The relevant set J_κ is the smallest set that contains $S_L \cup S_r$ and satisfies $S_j^\kappa \subseteq J_\kappa$ for every $j \in J_\kappa \cap \Delta_\kappa$. A set $K \subseteq [n]$ is κ -closed when $S_j^\kappa \subseteq K$ for every $j \in K \cap \Delta_\kappa$.*

The recursion is well defined because $S_i \subseteq [i-1]$; the smallest set exists by intersection of closed sets.

F.2 Assumptions and Their Deployment Reading

These are the assumptions for Proposition A.5.1 (Section 3.4).

- (G1) *Support.* For every $j \in J_\kappa \cap (\Delta_\kappa \setminus D_\kappa)$ and every $z \in \mathcal{V}_{S_j} \setminus \mathcal{K}_j$, $P^\kappa(V_{S_j} = z, R_t = r) > 0$ implies $P^0(V_{S_j} = z) > 0$.
- (G2) *Recorded downstream inputs.* $S_j \subseteq \mathcal{O}$ for every $j \in J_\kappa \cap (\Delta_\kappa \setminus D_\kappa)$.
- (G3) *Recorded loss inputs.* $S_L \subseteq \mathcal{O}$.
- (G4) *Recorded assignment.* $S_j^\kappa \subseteq \mathcal{O}$ and k_j^κ is known for every $j \in J_\kappa \cap D_\kappa$.

For an unchanged agent reading full text context, (G1) fails if a refusal message or regenerated text has zero baseline probability at that position. Coarsening that context creates support only under a sufficient-input assumption that logs alone cannot confirm.

F.3 Structural Lemmas

These lemmas support the proof of Proposition A.5.1 and its extension to unsupported inputs (Section 3.4).

Every $i \in \bar{\Delta}_\kappa$ has $i \notin D_\kappa$ and $S_i \subseteq \bar{\Delta}_\kappa$, so $\Delta_\kappa \subseteq \{s, \dots, n\}$ and $S_r \subseteq \bar{\Delta}_\kappa$; minimality and (G2) to (G4) give $J_\kappa \subseteq \mathcal{O}$.

Lemma F.3.1 (Truncated Factorization). *Under both laws, the marginal of $V_{\bar{\Delta}_\kappa}$ is $\prod_{i \in \bar{\Delta}_\kappa} k_i(v_i \mid v_{S_i})$; in particular P^κ and P^0 agree on $V_{\bar{\Delta}_\kappa}$ and on R_t . For every κ -closed K and $w \in \mathcal{V}_K$,*

$$\begin{aligned} P^\kappa(V_K = w) \\ = P^0(V_{K \cap \bar{\Delta}_\kappa} = w_{K \cap \bar{\Delta}_\kappa}) \\ \prod_{j \in K \cap \Delta_\kappa} k_j^\kappa(w_j \mid w_{S_j^\kappa}). \end{aligned} \quad (11)$$

Proof sketch. Apply the causal Markov factorization (Pearl, 2009a, Theorem 1, Eq. (15)) to the finite independent-seed model of P^κ , in whose replacement graph $\bar{\Delta}_\kappa$ and $K \cup \bar{\Delta}_\kappa$ are ancestral. Marginalizing the latter to K gives (11) by κ -closure. $\square$

Lemma F.3.2 (Kernel Recovery). *Let $j \in \Delta_\kappa$, and let $K \subseteq \bar{\Delta}_\kappa \cup (\Delta_\kappa \cap [j-1])$ contain S_j . If $P^0(V_K = u) > 0$, then $P^0(V_j = a \mid V_K = u) = k_j(a \mid u_{S_j})$ for every $a \in \mathcal{V}_j$, where k_j is the baseline kernel.*

Proof. Because K contains only nondescendants of j along with its inputs S_j , the claim follows from the parental Markov condition for (9) (Pearl, 2009b, Theorem 1.2.7). $\square$

F.4 Full Statement and Proof of

Proposition A.5.1

Enumerate $J_\kappa \cap \Delta_\kappa = \{j_1 < \dots < j_q\}$ and set $K_m = (J_\kappa \cap \Delta_\kappa) \cup \{j_1, \dots, j_m\}$ for $0 \leq m \leq q$ ($K_q = J_\kappa$). For unchanged j , let $\mathcal{Z}_j \subseteq \mathcal{V}_{S_j} \setminus \mathcal{K}_j$ be inputs, and let λ assign $\lambda_{j,z} \in \Delta(\mathcal{V}_j)$ to each $z \in \mathcal{Z}_j$. For r -compatible $v \in \mathcal{V}_{J_\kappa}$, define $G_0^\lambda(v) = P^0(V_{K_0} = v_{K_0})$ and $G_m^\lambda(v) = G_{m-1}^\lambda(v) c_{j_m}^\lambda(v)$ (0 if $G_{m-1}^\lambda(v) = 0$), where

$$c_j^\lambda(v) = \begin{cases} k_j(v_j | v_{S_j}), & j \notin D_\kappa, v_{S_j} \in \mathcal{K}_j, \\ P^0(V_j = v_j | V_{S_j} = v_{S_j}), & j \notin D_\kappa, v_{S_j} \notin \mathcal{K}_j \cup \mathcal{Z}_j, \\ k_j^\kappa(v_j | v_{S_j}^\kappa), & j \in D_\kappa, \\ \lambda_{j,v_{S_j}}(v_j), & j \notin D_\kappa, v_{S_j} \in \mathcal{Z}_j. \end{cases} \quad (12)$$

Here every $\mathcal{Z}_j = \emptyset$, so the fourth case never applies; write c_j and $G = G_q$. Section F.5 takes $\mathcal{Z}_j$ to be the null inputs.

Full Statement. In the finite coordinate model, let $P^0(R_t = r) > 0$.

- (i) *g-formula.* Under (G1), (G2), (G3), and (G4), every c_j used in G is well defined, and $P^\kappa(V_{J_\kappa} = v) = G(v)$ for every r -compatible $v \in \mathcal{V}_{J_\kappa}$. Hence, for every $B \subseteq \mathbb{L}$,

$$Q_t^\kappa(B | r) = \frac{1}{P^0(R_t = r)} \sum_{\substack{v \in \mathcal{V}_{J_\kappa}: \\ v_{S_r} = r, \\ \bar{\ell}(v_{S_L}) \in B}} G(v). \quad (13)$$

The right side depends only on the baseline record law, the declared inputs, the declared kernel values, $\bar{\ell}$, and the known replacement kernels. The vector law $Q_t^\kappa(\cdot | r)$ and every query $\rho(\phi_\# Q_t^\kappa(\cdot | r))$ are therefore identified within mechanisms that share the declared inputs and declared kernel values.

- (ii) *Support is checkable.* Under (G2), (G3), and (G4), whether (G1) holds is determined by the baseline record law and the declared kernel values.
- (iii) *Zero agreement.* There exists a mechanism satisfying (G1) to (G4) in which the baseline

and the replacement are deterministic given a recorded tape. Its agreement mass is 0, and its replacement risk is identified as $1/2$.

- (iv) *Composition with Proposition A.4.1.* Let the coordinate list embed a tree mechanism of Proposition A.4.1 as in Section F.1, with (R), (A1), (A2), and (A3) of that proposition, and a conditioning record generated before every replaced slot. Then (G1) holds at every input that the replacement reaches on an agreement path, and every downstream input reached after a terminal deviation is declared. Under (G1) to (G4), Proposition A.4.1(i) and (ii) give the restrictions of $Q_t^\kappa(\cdot | r)$ to the agreement set and to terminal deviations. The continuing part is therefore identified as the remainder.
- (v) *Unsupported inputs.* Without (G1), the identified set is the image described in Proposition F.5.2.

Conditions (G2) and (G4) cannot be dropped, and (G3) cannot be weakened to recording L alone (verify_supported.py, Appendix H).

Proof sketch. Part (i) is the sequential g-formula (Robins, 1986; Pearl and Robins, 1995), established here in Proposition F.5.2(a),(c). For (ii), $J_\kappa \subseteq \mathcal{O}$ makes the recursion observable up to any first reached undeclared null input, which detects failure of (G1). For (iii), record all coordinates of $U \sim \text{Bernoulli}(1/2)$, baseline $A = U$, replacement $A = 1 - U$, and Y whose kernel reads only A , with $P(Y = 1 | A = 0) = 1/4$ and $P(Y = 1 | A = 1) = 3/4$; take $L = Y$ and no conditioning record. Both outcome inputs have baseline support, all decisions disagree, and the g-formula gives replacement risk $1/2$. For (iv), Lemma E.3.1 gives support on agreement paths, and (A3) declares terminal valuations; subtract the two restrictions in Proposition A.4.1. Part (v) is Proposition F.5.2. $\square$

F.5 Continuation from Records with Unsupported Inputs

To handle unsupported inputs in Section 3.4, we keep (A1), (A2), and (G2) to (G4), drop (G1), and vary kernel laws at unsupported inputs within allowed sets.

In the coordinate model of Section F.1, let $\mathcal{U}_\kappa = J_\kappa \cap (\Delta_\kappa \setminus D_\kappa)$. For $j \in \mathcal{U}_\kappa$ and $z \in \mathcal{V}_{S_j} \setminus \mathcal{K}_j$, kernel j emits values in a nonempty allowed set

$\mathcal{E}_{j,z} \subseteq \mathcal{V}_j$ (default $\mathcal{V}_j$). Members of $\mathcal{M}_0$ share the declared inputs, declared kernel values, known replacement kernels, and allowed sets. Each coordinate has its own kernel; restrictions tying slots are addressed in Remark F.5.4. The population identified set is

$$\Theta_r = \{Q_{t,M}^\kappa(\cdot | r) : M \in \mathcal{M}_0, V_{\mathcal{O}} \text{ has the observed record law under } P_M^0\}. \quad (14)$$

Under (G2), an input $z \in \mathcal{V}_{S_j} \setminus \mathcal{K}_j$ with $P^0(V_{S_j} = z) = 0$ is a null input; collect them in $\mathcal{Z}_j$. The null blocks are $\mathcal{B} = \{(j, z) : j \in \mathcal{U}_\kappa, z \in \mathcal{Z}_j\}$, and free families form $\Lambda = \prod_{(j,z) \in \mathcal{B}} \Delta(\mathcal{E}_{j,z})$. For a member M with the observed record law, define its own null-input family by $\lambda_M = (k_j^M(\cdot | z))_{(j,z) \in \mathcal{B}}$. For $\lambda \in \Lambda$, the recursion of Section F.4 defines G^λ via (12), and

$$Q^\lambda(B | r) = \frac{1}{P^0(R_t = r)} \sum_{\substack{v \in \mathcal{V}_{J_\kappa}: \\ v_{S_r} = r, \\ \bar{\ell}(v_{S_L}) \in B}} G^\lambda(v). \quad (15)$$

The class $\mathcal{M}_0$ is *closed at null inputs* if for each member M with the observed record law and each $\lambda \in \Lambda$, setting $k_j(\cdot | z) = \lambda_{j,z}$ on $\mathcal{B}$ yields a member $M^\lambda \in \mathcal{M}_0$.

Lemma F.5.1 (Zero-Support Modification in the Coordinate Model). *Let M and M' have the same declared inputs and the same kernels, except kernel j at inputs in a set $\mathcal{Z} \subseteq \mathcal{V}_{S_j}$. If $P_M^0(V_{S_j} \in \mathcal{Z}) = 0$, then $P_{M'}^0 = P_M^0$.*

Proof. The products (9) of M and M' differ only where $v_{S_j} \in \mathcal{Z}$. Since $S_j \subseteq [j-1]$, the probability of $\{V_{S_j} \in \mathcal{Z}\}$ depends only on the shared kernels $1, \dots, j-1$, so it is 0 under both and both laws vanish there. $\square$

Proposition F.5.2 (Continuation from Records with Unsupported Inputs). *In the finite coordinate model, assume (A1) and (A2) in the form (10), together with (G2), (G3), and (G4). Condition (G1) is not assumed.*

(a) *Outer set. Let M have the declared inputs, the declared kernel values, and the known replacement kernels, with $P_M^0(R_t = r) > 0$. Formula (15), computed from the record law of M with $\lambda = \lambda_M$, equals $Q_{t,M}^\kappa(\cdot | r)$. Hence $\Theta_r \subseteq \{Q^\lambda(\cdot | r) : \lambda \in \Lambda\}$, where each Q^λ is computed from the observed record law.*

(b) *Sharpness. If $\mathcal{M}_0$ is closed at null inputs, then $\Theta_r = \{Q^\lambda(\cdot | r) : \lambda \in \Lambda\}$.*

(c) *Supported extreme. Suppose (G1) holds; by Proposition A.5.1(ii), this is a property of the observed record law and the declared kernel values. Then $Q^\lambda(\cdot | r)$ equals the right side of (13) for every $\lambda \in \Lambda$, and the image is the single law of Proposition A.5.1(i).*

Proof of (a). Fix M and $\lambda = \lambda_M$. Each K_m is κ -closed, so Lemma F.3.1 gives

$$\begin{aligned} \Pi_m(v) &:= P_M^\kappa(V_{K_m} = v_{K_m}) \\ &= P_M^0(V_{K_0} = v_{K_0}) \\ &\quad \prod_{l \leq m} k_{j_l}^\kappa(v_{j_l} | v_{S_{j_l}}^\kappa). \end{aligned} \quad (16)$$

Lemma F.3.2 and the definitions of declared, replacement, and null factors give $G_m^{\lambda_M} = \Pi_m$, including zero prefixes. Summing and conditioning on r gives (15); $J_\kappa \subseteq \mathcal{O}$ makes all terms except λ observable, proving the inclusion. $\square$

Proof of (b). For $\lambda \in \Lambda$ and a compatible M , closure gives $M^\lambda \in \mathcal{M}_0$. Resetting the null blocks one at a time, Lemma F.5.1 keeps the baseline law at each step, so $P_{M^\lambda}^0 = P_M^0$ and $\lambda_{M^\lambda} = \lambda$. Part (a) for M^λ gives $Q^\lambda(\cdot | r) = Q_{t,M^\lambda}^\kappa(\cdot | r) \in \Theta_r$. $\square$

Proof of (c). By (G1), no first undeclared null input can follow a positive r -compatible prefix in (a); zero prefixes stay zero. Thus no free factor is used, so $G^\lambda = G$ for every λ , giving (13) by (a). $\square$

Remark F.5.3 (What Finite Logs Show). Under (G2) to (G4), the population record law determines null inputs. Absence from a finite log does not prove an input is null. Treating unobserved inputs as null still yields a valid outer set. Setting free arguments to the true conditional law returns (12), so the enlarged image contains the true identified set. Finite-sample inference is a separate claim (Limitations).

Remark F.5.4 (Scope of Sharpness). Part (b) uses that each coordinate's kernel is free at its own null inputs. In the slot embedding of Section F.1, several slots can host one named kernel. A class that ties slots, such as one law per kernel and step when the embedding needs more slots than kernel and step pairs, is covered by the outer-set clause of Proposition A.5.1. It is a subclass with the same evidence, so part (a) still gives an outer set for it. Its identified set can be smaller.

F.6 Model Updates: Statement and Proof

This is the model-release result of Section 3.4.

Corollary F.6.1 (Model updates). *Assume (A1) and (A2) for every replaced decision, and let κ replace the agent policy at its decisions from the boundary onward, under an interface whose output contains every action and dynamic state that later kernels or ℓ read; a model update applied from the episode start replaces every decision. Suppose the baseline policy is deterministic given its decision input and a random tape, and that decision inputs, tapes, and baseline outputs are recorded. If the new policy is seed-addressable, that is, evaluable deterministically at each logged input and tape, then the agreement mass $m_\kappa(r) = P^\kappa(A_\kappa \mid r)$ is identified. It is estimated by the empirical agreement frequency over independent baseline episodes and is at most the first-decision agreement probability. With the loss value recorded, the agreement contribution is identified as well.*

Proof sketch. Including tapes in the inputs, the stated determinism and recording give (R); Lemma E.3.1 identifies agreement mass and its logged loss contribution. Seed-addressable evaluation determines the agreement indicators, whose empirical frequency is consistent by the strong law. The bound follows since agreement throughout implies first-decision agreement. $\square$

F.7 Budgeted Targeted Execution: Statement and Proof

This subsection proves the sampling and restoration bounds used in Section 3.4. Scripts in checks/ reproduce the E1 and E7 bounds and check finite instances; Appendix H and checks/README.md map each script to its claim.

Replay supplies the agree and stop terms of display (2); a uniform sample of unresolved continuations bounds the rest. Hypergeometric and binomial inversion (Clopper and Pearson, 1934), a union bound, and error spending over declared looks (Lan and DeMets, 1983) determine how many executions a decision needs.

Setting. Let harm be binary and q the changed system's $P(\text{harm})$, the target $q_t^\kappa(r)$ with X the harm indicator. A pass-all log holds n episodes. Under (R), (A1), (A2), and (A3), replay resolves an episode's outcome under the change when it agrees, when harm occurred before its first deviation, or when it stops and the abort valuation fixes the harm

indicator (a point mass, as when a stop ends the episode with the harm accrued before it); b resolved episodes are harmful. A nondegenerate abort valuation would need an independent draw for each stopped episode. The remaining N continuing episodes are unresolved, partitioned into strata s of sizes N_s fixed by the log (for example, by the step of first deviation). Executing an unresolved episode's continuation from its snapshot gives its outcome under the change, by (T). The completed cohort assigns each resolved episode its replayed outcome and each unresolved episode the outcome of its continuation; S is its harm count.

Hypothesis (H). The completed outcomes are independent Bernoulli variables with common probability q ; the random order in which unresolved episodes are executed is drawn independently of them, and the looks are fixed in advance. Independently sampled tasks and fresh seeds support (H); (T) links each continuation to its live law but does not by itself make outcomes independent across episodes, so dependence through shared serving or environment state lies outside it.

Bounds. In stratum s , execute a uniform random sample without replacement of k_s unresolved episodes and observe x_s harms. With $H(N_s, D, k_s)$ the hypergeometric law of harms in such a sample when D of the N_s are harmful, let

$$\begin{aligned} D_s^U(x_s, k_s; e) \\ = \max \{ D \in \{x_s, \dots, N_s - k_s + x_s\} : \\ \Pr_{H(N_s, D, k_s)}(X \leq x_s) > e \}, \end{aligned}$$

with $D_s^U = N_s$ when $k_s = 0$; at $k_s = N_s$ it returns x_s . Let $S^U = b + \sum_s D_s^U$ and let $u_{CP}(z, n; a)$ be the one-sided Clopper-Pearson upper limit for z successes in n trials at level $1 - a$, with $u_{CP}(n, n; a) = 1$.

Proposition F.7.1 (Budgeted targeted execution).

(i) *Given the completed cohort, $\Pr(S \leq S^U) \geq 1 - \sum_s e_s$, where stratum s uses error e_s .* (ii) *Under (H), $\Pr(q \leq u_{CP}(S^U, n; a_L)) \geq 1 - a_L - \sum_s e_s$.* (iii) *Let samples be nested over declared looks $j = 1, \dots, J$ with sizes $k_{s,j}$ and errors $e_{s,j}$, $\sum_{s,j} e_{s,j} \leq a_E$, and let $\bar{U}_j = \min_{l \leq j} u_{CP}(S_l^U, n; a_L)$. Under (H), $\Pr(q \leq \bar{U}_j \text{ for all } j) \geq 1 - a_L - a_E$. Stopping at the first look whose bound clears a target, or at look J , keeps this coverage, for every target at once.*

Proof. (i) If $D > D_s^U(x_s, k_s; e_s)$ then $\Pr_{H(N_s, D, k_s)}(X \leq x_s) \leq e_s$, so in each stratum the failure event is a lower tail of probability at most e_s ; a union bound gives the claim. (ii) Combine (i) with Clopper-Pearson coverage for $S \sim \text{Bin}(n, q)$ under (H) and monotonicity in the count (Clopper and Pearson, 1934). (iii) A union bound over strata and declared looks, intersected with the single Clopper-Pearson event for S , gives simultaneous coverage of the running minima. This also covers any selected look and every target. $\square$

The guarantees above concern sampling error under exact restoration (T). Lemma F.7.2 prices restoration error, and Remark F.7.3 incorporates it into certified bounds.

Lemma F.7.2 (Restoration Error). *Assume (R) and (A1) to (A3), and let ν , w_h , and $Q_h = \phi_{\#} P^{\kappa}(L \in \cdot \mid [h])$ be as in Proposition E.4.3. Let $\hat{Q}_h$ be the law of $X = \phi(L)$ in the run restored at h^+ , so that (T) says $\hat{Q}_h = Q_h$, and let $\gamma_h = \max_B |\hat{Q}_h(B) - Q_h(B)|$. Let $\hat{P} = \nu + \sum_h w_h \hat{Q}_h$ be the law that display (2) gives when its continue term uses the restored laws, with mean $\hat{\mu}$, and let $\varepsilon_{\kappa} = \sum_h w_h$ be the continuing mass. Then for every event B of X ,*

$$\begin{aligned} & |\hat{P}(B) - P^{\kappa}(X \in B \mid r)| \\ & \leq \sum_h w_h \gamma_h \leq \varepsilon_{\kappa} \max_h \gamma_h, \\ & |\hat{\mu} - E^{\kappa}[X \mid r]| \\ & \leq (\max \mathcal{X} - \min \mathcal{X}) \sum_h w_h \gamma_h. \end{aligned}$$

Proof. Both event bounds follow from the decomposition in Proposition E.4.3 and the triangle inequality; integrating tail differences over $[\min \mathcal{X}, \max \mathcal{X}]$ gives the mean bound. $\square$

Remark F.7.3 (Restoration Allowance). Proposition F.7.1 bounds the harm probability $\hat{q}$ of $\hat{P}$ when the completed restored outcomes satisfy hypothesis (H): independent Bernoulli with a common probability. Let u be its certified bound and $\xi = \sum_h w_h \gamma_h$. Lemma F.7.2 gives $|q - \hat{q}| \leq \xi$, so with the probability of the proposition $q \leq u + \xi$, and a target q^* is cleared when $u + \xi \leq q^*$; exact (T) is the case $\xi = 0$. For a tolerance $\xi_0 \geq \xi$ declared in advance, stopping at the first look with $\bar{U}_j + \xi_0 \leq q^*$ keeps the coverage of part (iii). The continuing mass ε_{κ} is estimated from the log, so

a population statement uses an upper confidence limit for it, whose error adds to that of the proposition, or the bound $\varepsilon_{\kappa} \leq 1$. At prefixes with harm before the deviation, $Q_h = \hat{Q}_h$ for harm, so they drop out of the harm allowance; task success can still depend on the continuation, so a decision with a success floor needs its own allowance from the same lemma. Both bounds of the lemma are attained: with $Q_h = \delta_{\min \mathcal{X}}$ and $\hat{Q}_h = (1 - \gamma)\delta_{\min \mathcal{X}} + \gamma\delta_{\max \mathcal{X}}$ for every h , the errors equal $\varepsilon_{\kappa}\gamma$ and $(\max \mathcal{X} - \min \mathcal{X})\varepsilon_{\kappa}\gamma$. A paired live run measures $q - \hat{q}$ itself and does not bound γ_h , since errors at different prefixes can offset.

F.8 Necessity of the Execution Budget

This subsection proves the execution limits and achievability claims of Section 3.4.

Setting. Let harm be binary, and assume (R), (A1) to (A3), and (S) with feasible set $\{0, 1\}$ at every unresolved prefix. Episodes are independent. Each is resolved and harmful with probability q_0 , unresolved with probability $m > 0$, and resolved and harmless otherwise. An unresolved continuation is harmful with probability $p \in (0, 1)$. Write $q = q_0 + mp$, and assume $\zeta = q^* - q > 0$ and $q^* < q_0 + m$. A world gives every unresolved prefix a live harm probability p_0 and a restored one $\hat{p}$ with $|p_0 - \hat{p}| \leq \gamma$. Condition (S) frees every live p_0 for a mechanism with the observed baseline law. We also let every restored $\hat{p}$ within γ of p_0 occur, as in the attainment part of Remark F.7.3. The log then has one law across worlds, and each restored cohort satisfies (H) because episodes are independent. A procedure reads the log, may randomize internally, and executes unresolved episodes one at a time, each at most once and chosen adaptively. An execution reveals the record of one restored continuation, independent of the other executions given the log. Worlds may give these records any law with the stated harm probabilities, so a world can change a harm probability and keep the conditional law of the record given harm. It stops after $K \leq n$ executions and certifies or not. Fix $\alpha, \beta \in (0, 1)$ with $\alpha + \beta < 1$ in this subsection; probabilities average over the log and any randomization. The size of a procedure is its largest certification probability over worlds with $q_0 + mp_0 > q^*$. Its power is its certification probability in the world W_1 with $p_0 = \hat{p} = p$. It is valid if its size is at most α and powerful if its power is at least $1 - \beta$. Let kl be the binary relative entropy and $\bar{p} = p + (\zeta - m\gamma)/m$.

Proposition F.8.1 (Execution Budget: Necessity). *In the setting above, the following hold.*

- (i) *If $m\gamma \geq \zeta$, every procedure has power at most its size, so no valid procedure is powerful, for any n .*
- (ii) *If $m\gamma < \zeta$, every valid and powerful procedure has $\mathbb{E}_{W_1}[K] \geq \text{kl}(1 - \beta, \alpha) / \text{kl}(p, \bar{p}) \geq \text{kl}(1 - \beta, \alpha) m^2 \bar{p}(1 - \bar{p}) / (\zeta - m\gamma)^2$. The first bound diverges as $m\gamma \uparrow \zeta$.*
- (iii) *Consider a sequential test of independent live episodes whose stopping time K_L has finite mean at live harm q . If it certifies with probability at most α under every episode law whose harm probability exceeds q^* , and at least $1 - \beta$ at harm q , then $\mathbb{E}_q[K_L] \geq \text{kl}(1 - \beta, \alpha) / \text{kl}(q, q^*) \geq \text{kl}(1 - \beta, \alpha) q^*(1 - q^*) / \zeta^2$.*

Proof. For $\varepsilon > 0$, let W_0^ε have $p_0 = p + (\zeta + \varepsilon)/m$ and $\hat{p}_\varepsilon = \max\{p_0 - \gamma, p\}$. For small ε it is a world, since $p_0 < 1$ when $q^* < q_0 + m$, and its live harm is $q^* + \varepsilon$. At every prefix, W_0^ε keeps the conditional law of the restored record given harm from W_1 . Let E be the event that the procedure certifies, with laws $\mathbb{P}_1$ in W_1 and $\mathbb{P}_0$ in W_0^ε . The log has the same law in both. Since W_0^ε has live harm above q^* , $\mathbb{P}_0(E)$ is at most the size, and at most α for a valid procedure.

(i) Let $m\gamma \geq \zeta$. Then $\hat{p}_\varepsilon - p \leq \varepsilon/m$. Draw one uniform variable per unresolved episode, and let the episode be harmful when the variable falls below p in W_1 and below $\hat{p}_\varepsilon$ in W_0^ε . Draw the rest of its record from the common conditional law given harm. The two outcomes differ with probability at most ε/m per episode. The procedure acts identically unless one of the at most n unresolved outcomes differs, so $\mathbb{P}_1(E) \leq \mathbb{P}_0(E) + n\varepsilon/m$. Letting $\varepsilon \downarrow 0$ bounds the power by the size. A valid procedure has size at most $\alpha < 1 - \beta$, so it is not powerful.

(ii) Let $m\gamma < \zeta$, so $\hat{p}_\varepsilon = \bar{p} + \varepsilon/m \in (p, 1)$ for small ε , since $\bar{p} \leq (q^* - q_0)/m < 1$. Conditional on the common log and internal randomization $\mathcal{G}$, unused episodes are independent arms. Their record laws have common conditional laws given harm, hence are mutually absolutely continuous with divergence $\text{kl}(p, \hat{p}_\varepsilon)$. The stopping time satisfies $K \leq n$. Apply Kaufmann et al. (2016, Lemma 1) conditionally and average using joint

convexity of kl :

$$\begin{aligned} \mathbb{E}_1[K] \text{kl}(p, \hat{p}_\varepsilon) &\geq \text{kl}(\mathbb{P}_1(E), \mathbb{P}_0(E)) \\ &\geq \text{kl}(1 - \beta, \alpha), \end{aligned}$$

by validity, power, and binary-KL monotonicity. Let $\varepsilon \downarrow 0$. The standard binary-KL bound $\text{kl}(p, \bar{p}) \leq (\bar{p} - p)^2 / \{\bar{p}(1 - \bar{p})\}$ gives the second bound. As $m\gamma \uparrow \zeta$, $\bar{p} \downarrow p$ and the first bound diverges.

(iii) Give the unsafe live law harm probability $q^* + \varepsilon$, keeping the conditional record law given harm from W_1 . For small ε , $0 < q < q^* + \varepsilon < 1$. Apply the same lemma with one arm, stopping at $\min\{K_L, t\}$ and event $E \cap \{K_L \leq t\}$. Let $t \rightarrow \infty$, using $\mathbb{E}_q[K_L] < \infty$, and then $\varepsilon \downarrow 0$. The same binary-KL bound gives the second inequality. $\square$

Achievability. Sequential probability ratio tests (SPRTs; Wald, 1945) approach the bounds of Proposition F.8.1(ii) and (iii) as error levels fall and the log grows, with a finite-pool allowance; Remark F.8.4 treats estimation of q_0 and m .

Let $m\gamma < \zeta$, and put $\text{up}(a, c) = \log\{(1 - a)/(1 - c)\}$ for $0 < a < c < 1$. The log has one law in every world, so a procedure may use q_0 and m . The *execution test* with levels $\alpha, \beta' \in (0, 1)$ uses $\bar{p} = (q^* - q_0)/m - \gamma$. It executes unresolved episodes in any order and reads only their harm bits. Starting at $\Xi_0 = 0$, it adds $\log(p/\bar{p})$ after a harm and $\text{up}(p, \bar{p})$ after a harmless outcome. It certifies when $\Xi_t \geq \log(1/\alpha)$, and stops without certifying when $\Xi_t \leq -\log(1/\beta')$ or all N unresolved episodes are used. The *live test* applies the same rule to live harm bits, with (q, q^*) in place of $(p, \bar{p})$ and no pool. A test with thresholds $A, B > 0$ uses A and B in place of $\log(1/\alpha)$ and $\log(1/\beta')$. Both tests look after every outcome, and Ville's inequality covers all looks.

Proposition F.8.2 (Execution Budget: Achievability). *Assume the setting of Proposition F.8.1 with $m\gamma < \zeta$. Let τ be the stopping time of the execution test with levels $\alpha, \beta' \in (0, 1)$ and an unlimited pool. Put $\bar{K}_{\text{ex}} = \{\log(1/\alpha) + \text{up}(p, \bar{p})\} / \text{kl}(p, \bar{p})$ and $\chi_n = 2\bar{K}_{\text{ex}} / (mn) + e^{-m^2 n/2}$.*

- (i) *The size of the test is at most α . For any thresholds, certification is nonincreasing in the restored harm rate, so the size is the certification probability at $\hat{p} = \bar{p}$. This holds in every world with $q_0 + mp_0 > q^*$, for every $\hat{p}$ within γ of p_0 and every record law given harm.*

(ii) The power is at least $1 - \beta' - \chi_n$. Also $\mathbb{E}_{W_1}[K] \leq \mathbb{E}_p[\tau] \leq \bar{K}_{\text{ex}}$, where $\mathbb{E}_p$ refers to independent Bernoulli(p) bits.

(iii) Let the live episodes be independent. The live test certifies with probability at most α when every episode has harm probability at least q^* , and with probability at least $1 - \beta'$ at harm q . Also $\mathbb{E}_q[K_L] \leq \bar{K}_{\text{lv}} = \{\log(1/\alpha) + \text{up}(q, q^*)\}/\text{kl}(q, q^*)$.

Proof. For the Bernoulli SPRT score Z (Wald, 1945), a direct computation gives $\mathbb{E}_\vartheta e^Z = 1 - (\vartheta - c)(c - a)/\{c(1 - c)\}$, $\mathbb{E}_a e^{-Z} = 1$, and $\mathbb{E}_a Z = \text{kl}(a, c)$ for $0 < a < c < 1$; take $(a, c) = (p, \bar{p})$.

(i) Given the log, executed bits are independent Bernoulli($\hat{p}$) with $\hat{p} \geq p_0 - \gamma > \bar{p}$, whatever the record law. Thus e^{Ξ_t} , stopped at the pool, is a non-negative supermartingale starting at one. Ville's inequality (Ville, 1939, Chapter V, Theorem 1, Eq. (15)), applied after adding the predictable decreases and approaching the threshold from below, bounds certification by α . Coupling by uniforms makes certification nonincreasing in $\hat{p}$; finite-pool continuity gives the size at $\bar{p}$.

(ii) At W_1 , Ville's inequality for $e^{-\Xi_t}$ bounds lower exit by β' . A run of $\lceil \{\log(1/\alpha) + \log(1/\beta')\}/\text{up}(p, \bar{p}) \rceil$ harmless outcomes leaves the strip, so τ has a geometric tail. Wald's identity (Wald, 1944), with upper overshoot at most $\text{up}(p, \bar{p})$, gives $\mathbb{E}_p[\tau] \leq \bar{K}_{\text{ex}}$. Since $K = \min\{\tau, N\} \leq \tau$, the expected-cost bound follows. Failure requires a lower exit or pool exhaustion; Markov's and Hoeffding's inequalities with $N \sim \text{Bin}(n, m)$ give $\Pr(\tau > N) \leq \chi_n$, hence power at least $1 - \beta' - \chi_n$.

(iii) The same argument with $(a, c) = (q, q^*)$ and no pool gives the live bounds. Size requires only conditional harm probability at least q^* ; independent Bernoulli(q) bits give the power and mean bounds. $\square$

Corollary F.8.3 (Cost Bounds and Ratio). *Let $m\gamma < \zeta$, $\alpha + \beta < 1$, and $\chi_n < \beta$. Let C_{ex} be the infimum of $\mathbb{E}_{W_1}[K]$ over valid and powerful procedures. Let C_{lv} be the infimum of $\mathbb{E}_q[K_L]$ over the live tests of Proposition F.8.1(iii). Write $\text{kl}_{\text{ex}} = \text{kl}(p, \bar{p})$, $\text{kl}_{\text{lv}} = \text{kl}(q, q^*)$, $\text{up}_{\text{ex}} = \text{up}(p, \bar{p})$,*

and $\text{up}_{\text{lv}} = \text{up}(q, q^)$. Then*

$$\frac{\text{kl}(1 - \beta, \alpha)}{\text{kl}_{\text{ex}}} \leq C_{\text{ex}} \leq \frac{\log(1/\alpha) + \text{up}_{\text{ex}}}{\text{kl}_{\text{ex}}},$$

$$\frac{\text{kl}(1 - \beta, \alpha)}{\text{kl}_{\text{lv}}} \leq C_{\text{lv}} \leq \frac{\log(1/\alpha) + \text{up}_{\text{lv}}}{\text{kl}_{\text{lv}}}.$$

Each upper bound exceeds its lower bound by at most $[\text{up} + \beta\{\log(1/\alpha) + \log(1/\beta) + 1\}]/\text{kl}$ with matching subscripts. Moreover

$$\frac{C_{\text{ex}}}{C_{\text{lv}}} \geq \frac{\text{kl}(1 - \beta, \alpha)}{\log(1/\alpha) + \text{up}_{\text{lv}}} \cdot \frac{\text{kl}_{\text{lv}}}{\text{kl}_{\text{ex}}},$$

$$\frac{C_{\text{ex}}}{C_{\text{lv}}} \leq \frac{\log(1/\alpha) + \text{up}_{\text{ex}}}{\text{kl}(1 - \beta, \alpha)} \cdot \frac{\text{kl}_{\text{lv}}}{\text{kl}_{\text{ex}}}.$$

Both outer factors tend to 1 as $\alpha, \beta \downarrow 0$ with n large enough for $\chi_n < \beta$, so $C_{\text{ex}}/C_{\text{lv}} \rightarrow \text{kl}_{\text{lv}}/\text{kl}_{\text{ex}}$.

Proof. Combine Propositions F.8.1 and F.8.2, taking $\beta' = \beta - \chi_n$ for executions and $\beta' = \beta$ for live episodes; the ratio bounds divide the cost bounds, and expanding $\text{kl}(1 - \beta, \alpha)$ with $-(1 - \beta)\log(1 - \beta) \leq \beta$ and $\beta\log(1 - \alpha) \leq 0$ gives the gaps and limiting factors. $\square$

With Proposition F.8.1(i), this makes $\gamma = \zeta/m$ the sharp threshold as n grows. For every $\gamma < \zeta/m$ there is an n_0 , depending also on α and β , such that for $n \geq n_0$ a valid and powerful procedure exists. For $\gamma \geq \zeta/m$ none exists at any n .

Remark F.8.4 (Estimating q_0 and m). A finite log gives only b/n and N/n , and $\bar{p}$ decreases in q_0 and in m . Let $q_0^U \geq q_0$ and $m^U \geq m$ be Clopper-Pearson upper limits at level $\alpha_{\log}/2$ each, so that both hold with probability at least $1 - \alpha_{\log}$. Run the execution test with $\bar{p}^U = (q^* - q_0^U)/m^U - \gamma$ in place of $\bar{p}$ and the realized N as pool, and require $p < \bar{p}^U < 1$ before evaluating the test, never certifying when this condition fails. On the event of the limits, $\bar{p}^U \leq \bar{p} < \hat{p}$, so the size is at most $\alpha + \alpha_{\log}$. Given a log with $p < \bar{p}^U < 1$, the argument of (ii) at $\bar{p}^U$ gives $\mathbb{E}[K \mid \log] \leq \bar{K}^U$, and Markov's inequality bounds pool exhaustion by $\bar{K}^U/N$, so the failure probability is at most $\beta' + \bar{K}^U/N$, where $\bar{K}^U$ is $\bar{K}_{\text{ex}}$ at $\bar{p}^U$. Unconditionally, the power loses at most $\mathbb{P}(\bar{p}^U \leq p \text{ or } \bar{p}^U \geq 1) + \mathbb{E}[\min\{1, \bar{K}^U/N\}; p < \bar{p}^U < 1]$. Both terms vanish as n grows, because $\bar{p}^U \rightarrow \bar{p} \in (p, 1)$ and $N \rightarrow \infty$ in probability.

F.9 Execution Across Strata

This extends the execution bounds of Section 3.4 to strata.

Setting. We extend the framework of Appendix F.8 to strata $s = 1, \dots, S$, such as the step where the gate first blocks. An episode is unresolved in stratum s with probability $w_s > 0$, so $\sum_s w_s = m$. In stratum s , continuations are harmful with probability $p_s \in (0, 1)$, restoration error is at most $\gamma_s \in [0, 1]$, and each execution costs $c_s > 0$ (for example, in agent calls); live episodes cost $c_L > 0$. A world assigns stratum s live harm probability p_{0s} and restored rate $\hat{p}_s$ with $|p_{0s} - \hat{p}_s| \leq \gamma_s$. An adaptive procedure executes unused unresolved episodes in chosen strata, incurring cost $C = \sum_s c_s K_s$. Let $q = q_0 + \sum_s w_s p_s$, $\zeta = q^* - q > 0$, and $q^* < q_0 + m$. Power is evaluated at world W_1 ($p_{0s} = \hat{p}_s = p_s$), and size is the maximum certification probability over worlds with $q_0 + \sum_s w_s p_{0s} > q^*$.

A restored vector can come from a world of the null only if the live harm it allows, at most $q_0 + \sum_s w_s \min\{1, \hat{p}_s + \gamma_s\}$, exceeds q^* . So the restored vectors of the null lie in the convex set

$$\mathcal{A} = \left\{ r \in [0, 1]^S : q_0 + \sum_s w_s \min\{1, r_s + \gamma_s\} \geq q^* \right\},$$

and every $r \in (0, 1)^S$ with strict inequality is one. Let $\xi = \sum_s w_s \min\{\gamma_s, 1 - p_s\}$ cap each restoration error at $1 - p_s$ (Remark F.7.3), put $\zeta' = \zeta - \xi$, and let $J = \{s : p_s + \gamma_s < 1\}$. For $s \notin J$, null worlds can keep restored rate p_s while setting live harm to one, so the design below does not sample these strata. For $s \in J$ and $t \geq 0$, let $z_s(t) \in [p_s, 1)$ solve $\text{kl}(p_s, z_s(t)) = c_s t$, and set $b_s(t) = \min\{1 - \gamma_s, z_s(t)\}$. When $\zeta' > 0$, let t_* be the unique positive root of

$$\sum_{s \in J} w_s \{b_s(t) - p_s\} = \zeta', \quad (17)$$

We define the *design point*: $\bar{p}_s = b_s(t_*)$ on J and $\bar{p}_s = p_s$ off J ; active strata $J_* = \{s \in J : \bar{p}_s < 1 - \gamma_s\}$; slopes $g_s = (\bar{p}_s - p_s) / \{\bar{p}_s(1 - \bar{p}_s)\}$; execution shares $a_s \propto w_s / g_s$ on J_* ($a_s = 0$ elsewhere); and mean cost $\bar{c} = \sum_s a_s c_s$. With $S = 1$, $\bar{p}_1$ is the $\bar{p}$ of Appendix F.8.

Proposition F.9.1 (Execution across Strata: Necessity and Allocation). *In the setting above, the following hold.*

- (i) *If $\xi \geq \zeta$, every procedure has power at most its size. If every $p_s + \gamma_s \leq 1$, then $\xi = \sum_s w_s \gamma_s$; otherwise $\sum_s w_s \gamma_s \geq \zeta$ does not suffice.*

- (ii) *If $\zeta' > 0$, then (17) has a unique root $t_* > 0$, J_* is nonempty, and every valid and powerful procedure has $\mathbb{E}_{W_1}[C] \geq \text{kl}(1 - \beta, \alpha) / t_*$.*

- (iii) *Let $\zeta' > 0$. Then $t_* = \max_\lambda \inf_{r \in \mathcal{A}} \sum_s \lambda_s \text{kl}(p_s, r_s) / c_s = \min_{r \in \mathcal{A}} \max_s \text{kl}(p_s, r_s) / c_s$, with λ ranging over cost shares ($\lambda_s \geq 0$, $\sum_s \lambda_s = 1$). The cost shares $\lambda_s^* = a_s c_s / \bar{c}$ and the vector $\bar{p}$ form a saddle point, every stratum in J_* has $\text{kl}(p_s, \bar{p}_s) = c_s t_*$, and every $r \in \mathcal{A}$ has $\sum_{s \in J_*} w_s (r_s - \bar{p}_s) \geq 0$.*

- (iv) *Let every $p_s + \gamma_s < 1$ and $q^* \downarrow q + \xi$, and put $V = \sum_s w_s \sqrt{p_s(1 - p_s)c_s}$. Then $1/t_* = 2V^2 \zeta'^{-2} \{1 + O(\zeta')\}$, and a_s tends to $w_s \sqrt{p_s(1 - p_s)} / c_s / \sum_j w_j \sqrt{p_j(1 - p_j)} / c_j$, the cost-weighted allocation of Neyman (1934). With $S = 1$ and $c_1 = 1$, (i) and (ii) are Proposition F.8.1(i) and the first bound of (ii). If $\sum_s w_s \gamma_s < \zeta$, then $1/t_* \leq c_{\text{pool}} / \text{kl}(p_{\text{pool}}, p_{\text{pool}} + (\zeta - \sum_s w_s \gamma_s) / m)$ with $p_{\text{pool}} = \sum_s w_s p_s / m$ and $c_{\text{pool}} = \sum_s w_s c_s / m$: the strata never raise the bound of Proposition F.8.1(ii) applied to the pooled rate at the mean cost.*

- (v) *Replace the strata of a set I by one stratum with mass $\sum_{s \in I} w_s$, the mass-weighted means of p_s and γ_s , and a cost at least the mass-weighted mean of c_s . Then t_* does not increase, where $t_* = 0$ when $\zeta' \leq 0$, the max-min value of (iii) in that case because then $p \in \mathcal{A}$.*

Proof. (i) Since $\min\{1, p_s + \gamma_s\} = p_s + \min\{\gamma_s, 1 - p_s\}$, the live harm allowed by $\hat{p} = p$ is $q + \xi$. If $\xi > \zeta$, the world with $\hat{p} = p$, $p_{0s} = \min\{1, p_s + \gamma_s\}$, and the record law of W_1 given harm is in the null and gives the log and every executed record the laws they have in W_1 , so the power is at most the size. If $\xi = \zeta$, then $q + \xi = q^* < q_0 + m$, so some stratum s' has $p_{s'} + \gamma_{s'} < 1$. For small $\varepsilon > 0$, raise only $\hat{p}_{s'}$ to $p_{s'} + \varepsilon$ and keep $p_{0s} = \min\{1, \hat{p}_s + \gamma_s\}$ and the record law of W_1 given harm; the live harm is $q^* + w_{s'} \varepsilon$. The coupling of Proposition F.8.1(i) changes at most n outcomes, each with probability at most ε , and $\varepsilon \downarrow 0$ bounds the power by the size. For the second claim, let $q_0 = 0$, $w = (1/2, 1/2)$, $p = (0.95, 0.05)$, $\gamma = (0.2, 0)$, and $q^* = 0.58$. Then $\sum_s w_s \gamma_s = 0.10 > \zeta = 0.08$, but $\xi = 0.025$, and every world of the null has $\hat{p}_2 > 0.16$ because

$p_{01} \leq 1$. With $n = 4,000$, a test that executes 1,000 episodes of stratum 2 when the log holds that many, and certifies when at most 105 are harmful, has size below $e^{-2 \cdot 1000 \cdot 0.055^2} < 0.0024$ and failure probability below $e^{-2 \cdot 1000 \cdot 0.056^2} + e^{-500} < 0.0024$ (Hoeffding).

(iii) and (ii) Setting $r_s = p_s$ off J and replacing r_s by $\min\{1 - \gamma_s, \max\{p_s, r_s\}\}$ on J keep r in $\mathcal{A}$ and raise no objective. So the inner problems run over $p_s \leq r_s \leq 1 - \gamma_s$ on J with $\sum_{s \in J} w_s(r_s - p_s) \geq \zeta'$. Each $b_s(t) - p_s$ is continuous and nondecreasing from 0, and strictly increasing until it reaches $1 - \gamma_s - p_s$. The total capacity $\sum_{s \in J} w_s(1 - \gamma_s - p_s)$ equals $q_0 + m - q - \xi$ and exceeds ζ' . So t_* exists, is unique, and leaves J_* nonempty, and $\bar{p}$ meets the constraint with equality. Let $r \in \mathcal{A}$. Strata outside J_* contribute w_s to the live harm at $\bar{p}$, the most they can, so $\sum_{s \in J_*} w_s[\min\{1, r_s + \gamma_s\} - (\bar{p}_s + \gamma_s)] \geq 0$, and the bracket is at most $r_s - \bar{p}_s$; this is the last claim of (iii). Every stratum has $\text{kl}(p_s, \bar{p}_s) \leq c_s t_*$, with equality on J_* , so $\max_s \text{kl}(p_s, \bar{p}_s)/c_s = t_*$ and every λ has inner value at most t_* . Convexity of $\text{kl}(p_s, \cdot)$ and $\lambda_s^* g_s/c_s = a_s g_s/\bar{c} \propto w_s$ on J_* give, for $r \in \mathcal{A}$,

$$\begin{aligned} & \sum_s \frac{\lambda_s^*}{c_s} \text{kl}(p_s, r_s) \\ & \geq \sum_{s \in J_*} \frac{a_s}{\bar{c}} \{c_s t_* + g_s(r_s - \bar{p}_s)\} \\ & \geq t_*, \end{aligned}$$

so λ^* has inner value t_* , and $\max_s \text{kl}(p_s, r_s)/c_s \geq t_*$ on $\mathcal{A}$. This proves (iii) without exchanging the maximum and the infimum. For (ii), take r with strict inequality in $\mathcal{A}$ and $0 < r_s < 1$, and the world of the null with $\hat{p} = r$, $p_{0s} = \min\{1, r_s + \gamma_s\}$, and the record law of W_1 given harm. An execution in stratum s contributes relative entropy $\text{kl}(p_s, r_s)$, and Lemma 1 of Kaufmann et al. (2016), applied as in Proposition F.8.1(ii), gives $\sum_s \mathbb{E}_{W_1}[K_s] \text{kl}(p_s, r_s) \geq \text{kl}(1 - \beta, \alpha)$. With $B = \mathbb{E}_{W_1}[C]$ and $\lambda_s = c_s \mathbb{E}_{W_1}[K_s]/B$, this reads $B \sum_s \lambda_s \text{kl}(p_s, r_s)/c_s \geq \text{kl}(1 - \beta, \alpha)$. Points $\bar{p} + \delta \mathbf{1}_{J_*}$ are such r for small $\delta > 0$, so letting $\delta \downarrow 0$ gives $B \max_s \text{kl}(p_s, \bar{p}_s)/c_s \geq \text{kl}(1 - \beta, \alpha)$, that is, $B t_* \geq \text{kl}(1 - \beta, \alpha)$.

(iv) For small ζ' , no coordinate reaches $1 - \gamma_s$; Taylor expansion of binary relative entropy in (17) and $a_s \propto w_s/g_s$ gives both limits. With $S = 1$ and $c_1 = 1$, the setting's $q^* < q_0 + m$ gives $\zeta < m(1 - p)$, so $\xi = m \min\{\gamma, 1 - p\} \geq \zeta$ exactly when $m\gamma \geq \zeta$. If $m\gamma < \zeta$, then $\xi = m\gamma$,

the root of (17) is $\bar{p}_1 = p + (\zeta - m\gamma)/m$, and $t_* = \text{kl}(p, \bar{p})$. For pooling, choose $\lambda_s \propto w_s c_s$ in (iii). Joint convexity gives $\sum_s (w_s/m) \text{kl}(p_s, r_s) \geq \text{kl}(p_{\text{pool}}, \sum_s w_s r_s/m)$, where feasibility ensures $\sum_s w_s r_s/m \geq p_{\text{pool}} + (\zeta - \sum_s w_s \gamma_s)/m > p_{\text{pool}}$; monotonicity gives the bound.

(v) The claim is immediate if the merged $t_* = 0$. Otherwise split the optimal share of merged stratum ι as $\lambda_s = \lambda'_\iota w_s c_s / \sum_{j \in I} w_j c_j$. Map feasible r to r' by mass-weighted averaging on I ; concavity of $\min\{1, \cdot\}$ preserves feasibility. Joint convexity and the cost condition give $\sum_{s \in I} \lambda_s \text{kl}(p_s, r_s)/c_s \geq \lambda'_\iota \text{kl}(p_\iota, r'_\iota)/c_\iota$, so (iii) bounds the split value below by the merged value. $\square$

Achievability. The stratified execution test with levels $\alpha, \beta' \in (0, 1)$ draws stratum $s \in J_*$ with probability a_s , independently of the past, executes an unused unresolved episode, and reads its harm bit Y . Like the test of Proposition F.8.2, it uses p and the log's masses. Starting at $\Xi_0 = 0$, each draw adds score $Y \log(p_s/\bar{p}_s) + (1 - Y) \text{up}(p_s, \bar{p}_s)$. The test certifies when $\Xi_t \geq \log(1/\alpha)$, and stops without certifying when $\Xi_t \leq -\log(1/\beta')$ or the drawn stratum pool is exhausted. Set $\bar{u} = \max_{s \in J_*} \text{up}(p_s, \bar{p}_s)$, $\bar{K} = \{\log(1/\alpha) + \bar{u}\}/(t_* \bar{c})$, and $\chi_n = \sum_{s \in J_*} \{2a_s \bar{K}/(w_s n) + e^{-w_s^2 n/2}\}$.

Proposition F.9.2 (Execution across Strata: Achievability and Live Crossover). Assume the setting of Proposition F.9.1 with $\zeta' > 0$.

- (i) For every log, every restored vector in $\mathcal{A}$, and every record law given harm, the test certifies with probability at most α . This also holds when the design point is computed from design values $\tilde{p} \in (0, 1)^S$ in place of p , provided $\zeta' > 0$ at $\tilde{p}$, since $\mathcal{A}$ does not involve p .
- (ii) With unlimited pools, at W_1 the test stops at the lower threshold with probability at most β' , uses $\mathbb{E}_{W_1}[\tau] \leq \bar{K}$ draws, and costs $\mathbb{E}_{W_1}[C] \leq \{\log(1/\alpha) + \bar{u}\}/t_*$. With the pools of a log of n episodes, its power is at least $1 - \beta' - \chi_n$ and the cost bound holds.
- (iii) Let C_{ex} be the least $\mathbb{E}_{W_1}[C]$ over valid and powerful procedures, and C_{lv} the least expected cost, at c_L per episode, of the live tests of Proposition F.8.1(iii). If $\chi_n < \beta$, then $\text{kl}(1 - \beta, \alpha)/t_* \leq C_{\text{ex}} \leq \{\log(1/\alpha) + \bar{u}\}/t_*$. As $\alpha, \beta \downarrow 0$ with $\chi_n < \beta$, $C_{\text{ex}}/C_{\text{lv}} \rightarrow \text{kl}(q, q^*)/(c_L t_*)$. So executing the unresolved continuations is cheaper to first order exactly

when $t_* > t_L = \text{kl}(q, q^*)/c_L$, that is, when the left side of (17) at t_L is below ζ' .

Proof. Use the score identities and SPRT argument of Proposition F.8.2, with $(a, c) = (p_s, \bar{p}_s)$ in stratum s .

(i) Given the log, past, and stratum s , the bit is $\text{Bernoulli}(\hat{p}_s)$ whatever its record law. For $\hat{p} \in \mathcal{A}$, Proposition F.9.1(iii) and $a_s g_s \propto w_s$ give $\mathbb{E}[e^{Z_t} \mid \text{past}] = 1 - \sum_{s \in J_*} a_s g_s (\hat{p}_s - \bar{p}_s) \leq 1$. Extend each pool by independent $\text{Bernoulli}(\hat{p}_s)$ bits; finite-pool certification requires the unlimited supermartingale e^{Ξ_t} to cross $1/\alpha$. Ville’s inequality gives size at most α , also for $\tilde{p}$, since the argument uses only the design point’s properties relative to $\mathcal{A}$.

(ii) At W_1 , the mean score is $t_* \bar{c}$ and upper overshoot is at most $\bar{u}$. The same SPRT argument gives lower-exit probability at most β' , $\mathbb{E}[\tau] \leq \bar{K}$, and mean cost $\bar{c} \mathbb{E}[\tau]$. Wald’s identity gives mean draws $a_s \mathbb{E}[\tau] \leq a_s \bar{K}$ in stratum s . The same Markov and Hoeffding bounds, summed over strata, bound exhaustion by χ_n ; together with lower exit this gives the power, while exhaustion only reduces cost.

(iii) Combine Proposition F.9.1(ii) with (i) and (ii), taking $\beta' = \beta - \chi_n$, for the execution bounds. The live bounds of Propositions F.8.1(iii) and F.8.2(iii), and $\text{kl}(1 - \beta, \alpha)/\log(1/\alpha) \rightarrow 1$, give the cost ratio. Strict increase of the left side of (17) on $[0, t_*]$ gives the crossover equivalence. $\square$

Remark F.9.3 (Allocation, Pools, and Snapshots). The shares a are part of the test’s validity, not only of its cost. In the instance of `checks/verify_budget_strata.py`, the same increments with draws in proportion to w_s have a per-draw mean of e^Z above one at a boundary point of $\mathcal{A}$, and their size there is about 0.15 at $\alpha = 0.05$. Design values $\tilde{p} \in (0, 1)^S$ from a pilot on separate episodes, such as smoothed proportions, keep the size at α by part (i), with no certification when $\zeta' \leq 0$ at $\tilde{p}$; we do not analyze the cost of such an adaptive rule (compare Garivier and Kaufmann, 2016). The bounds concern population harm under independent outcomes, with power at exact restoration; fixed-cohort harm fractions require a sampling design such as Proposition F.7.1. Power at every approximately restored law is not guaranteed. With q_0 and w estimated, replacing them in $\mathcal{A}$ by simultaneous upper confidence limits of level $\alpha_{\log}$ enlarges $\mathcal{A}$ on the event of the limits. Running the test only when $q_0^U + \sum_s w_s^U \min\{1, \tilde{p}_s + \gamma_s\} < q^* < q_0^U + \sum_s w_s^U$

for these limits q_0^U and w^U , and otherwise not certifying, keeps the size at most $\alpha + \alpha_{\log}$, as in Remark F.8.4.

Runs from an earlier snapshot re-execute the agreeing steps (Remark F.10.2). Suppose they reach the first-deviation strata of a set I with probabilities $w_s / \sum_{j \in I} w_j$ in every world, with the same restored harm laws and admissible rates; that their records can be simulated from the stratum and one restored harm bit by a randomization that does not depend on the world; and that they cost at least the mean cost of runs started at those deviations. Then data processing and the share splitting of Proposition F.9.1(v) show that restoring at the first deviations of Proposition F.10.1 never raises the bound of part (ii). The comparison does not cover earlier runs that observe more, restore differently, or route differently, and choosing coarser snapshots trades storage against the bound, which we do not solve.

Related Tests. With unit costs and no restoration error, $\mathcal{A}$ is a half-space of mean vectors, and the saddle point of Proposition F.9.1(iii) is the half-space case of Juneja and Krishnasamy (2019, Theorem 5), whose least favorable alternative also equalizes the divergences. Multi-fidelity best-arm identification pairs cheaper samples of known maximal bias with their costs (Poiani et al., 2024). Stratified sequential audits permit arbitrary stratum selection (Spertus and Stark, 2022; Spertus et al., 2024); this likelihood-ratio test requires its design shares.

F.10 Snapshot Design for a Family of Gates

This proves the snapshot-design claim of Section 3.4.

Setting and Evidence. We adopt the framework of Appendix E.2 with query $X = \phi(L)$ and pre-intervention record r , where f^0 executes each call. Decisions represent steps in the demonstrations; gates acting on each call treat calls as decisions (via the refined clock of Appendix A.1). Let $m(h) = m_{t(h)}$ be the score in $I(h)$, and $\bar{m}(h)$ the maximal score over previous decision histories $h' \prec h$ with $t(h') \geq s$ for boundary step s ($-\infty$ if none). The retry gate κ_θ of Section 2.2 replaces the call by a refusal $a_\theta(i) \neq f^0(i)$ when $t \geq s$ and $m(i) > \theta$. A refusal at final step T constitutes an abort, as no decision history follows (Appendix E.2). At each step $t < T$, a decision history follows the refusal.

Write $w_h = P^0([h] \mid r)$, $\mathcal{D}_\kappa^+ = \mathcal{D}_\kappa^{\text{cont}}(r, P^0)$, and $Q_h^\kappa = \phi_\# P^\kappa(L \in \cdot \mid [h])$ for $h \in \mathcal{D}_\kappa^+$. Feasible set $\mathcal{X}_h^\kappa$ follows Appendix E.1 for gate κ , de-

pending on κ via subsequent refusals. A history $h \in \mathcal{D}_\kappa^+$ is *open for κ* when $|\mathcal{X}_h^\kappa| \geq 2$ and closed otherwise (for instance, when X is the harm indicator and harm accrued before h). Let $\mathcal{O}_\kappa$ collect histories open for κ .

A set $\mathcal{S}$ of logged decision histories h with $[h] \subseteq \Omega_r$ and $P^0([h]) > 0$ stores *first-deviation snapshots*. For $h \in \mathcal{S}$ and gate κ with $h \in \mathcal{D}_\kappa^+$, the evaluator restores at h_κ^+ and executes the continuation. Assumption (T) equates this continuation law with Q_h^κ , estimated by finite executions (Proposition F.7.1). Earlier runs re-execute agreeing steps as partial live runs (Remark F.10.2). Condition (U) of Section 3.4 admits arbitrary response laws at inputs of zero baseline probability, taking model class $\mathcal{M}_{\text{full}}$ from Lemma E.4.2. Its mechanisms satisfy (R) and (A1) to (A3), with unreplaced kernels reading the full history, so (G1) fails at refusal inputs. The proof of (c) does not require realizability.

Proposition F.10.1 (Snapshots for a Threshold Family). *In (a) and (b), let h be a decision history with $s \leq t(h) < T$, $[h] \subseteq \Omega_r$, and $P^0([h]) > 0$.*

(a) *$h \in \mathcal{D}_{\kappa_\theta}^+$ exactly when $\bar{m}(h) \leq \theta < m(h)$, so h is a first deviation for some $\theta \in \Theta$ exactly when $\Theta \cap [\bar{m}(h), m(h)) \neq \emptyset$.*

(b) *For $\Theta = [\theta_{\min}, 1)$, this means $m(h) > \max\{\theta_{\min}, \bar{m}(h)\}$, a running maximum above $\theta_{\min}$, and the gates $\theta \in [\max\{\theta_{\min}, \bar{m}(h)\}, m(h))$ deviate at h . Ties and a score equal to $\theta_{\min}$ give no running maximum, and a score of 1 is one exactly when all earlier scores since s are below 1. For a finite $\Theta = \{\theta_1 < \dots < \theta_n\}$, the condition is $B(m(h)) > B(\bar{m}(h))$ with $B(x) = \#\{i : \theta_i < x\}$, and an episode has at most n first deviations.*

(c) *Assume (R), (A1) to (A3), (U), and (T). Given first-deviation snapshots $\mathcal{S}$, the identified set of $\phi_{\#} P^\kappa(L \in \cdot \mid r)$ is $\{\nu + \sum_{u \in \mathcal{D}_\kappa^+ \cap \mathcal{S}} w_u Q_u^\kappa + \sum_{u \in \mathcal{D}_\kappa^+ \setminus \mathcal{S}} w_u Q_u : Q_u \in \Delta(\mathcal{X}_u^\kappa)\}$ (ν as in Proposition E.4.3). It is a singleton exactly when $\mathcal{O}_\kappa \subseteq \mathcal{S}$; otherwise, for binary X , the set for $P^\kappa(X = 1 \mid r)$ has width $\sum_{u \in \mathcal{O}_\kappa \setminus \mathcal{S}} w_u$. So $\mathcal{S}$ identifies every κ_θ , $\theta \in \Theta$, exactly when it contains $\bigcup_\theta \mathcal{O}_{\kappa_\theta}$, the least such set. For $[\theta_{\min}, 1)$, these are the running maxima above $\theta_{\min}$ at steps $t < T$ that are open for some gate deviating there.*

Proof. (a) This is the first-deviation condition by definition, with continuation because $t(h) < T$.

(b) Intersecting $[\bar{m}(h), m(h))$ with Θ gives the stated cases. For finite Θ , $B(m) - B(\bar{m})$ counts

the available thresholds, and each threshold has at most one first deviation per episode.

(c) By Proposition E.4.3, a compatible model has law $\nu + \sum_u w_u Q_u^M$ with $Q_u^M \in \Delta(\mathcal{X}_u^\kappa)$, and (T) fixes $Q_u^M = Q_u^\kappa$ for $u \in \mathcal{D}_\kappa^+ \cap \mathcal{S}$. Conversely, let $M^* \in \mathcal{M}_{\text{full}}$ generate the evidence and fix $Q_u \in \Delta(\mathcal{X}_u^\kappa)$ for $u \in \mathcal{D}_\kappa^+ \setminus \mathcal{S}$. For each $x \in \mathcal{X}_u^\kappa$, take a replacement-consistent extension ω_x of u_κ^+ with $\phi(\ell(\omega_x)) = x$. By (U), set the kernels inside $[u_\kappa^+]$ to the conditional laws of $\sum_x Q_u(x) \delta_{\omega_x}$.

These cylinders are P^0 -null, so kernels set inside them do not enter the baseline law, which the modification therefore preserves. They are disjoint from $[g_{\kappa'}^+]$ for every gate κ' of the family and every $g \in \mathcal{S} \cap \mathcal{D}_{\kappa'}^+$ or terminal g of positive mass. If $u \neq g$ and they meet, one of u, g extends a deviating output of the other, which has baseline probability zero. A terminal $g = u$ has an abort output, whereas u_κ^+ continues. Thus the modification also preserves the stored laws and the valuations of (A3), and attains the chosen law. The set is a constant plus $w_u \Delta(\mathcal{X}_u^\kappa)$ for each $u \notin \mathcal{S}$, which gives the singleton criterion and the width. $\square$

Remark F.10.2 (Broader Evidence). The least-set claim concerns restoration at the first deviation of each gate. Suppose instead that $\mathcal{S}$ holds logged decision histories g with $[g] \subseteq \Omega_r$ and $P^0([g]) > 0$. Assume that (T) links a run of κ from g to the changed system only for gates that reach g , those that agree with f^0 at every earlier decision. Such a run re-executes the agreeing steps below g , and its law determines only the sum $\sum_{h \geq g} w_h Q_h^\kappa$ over the first deviations h of κ below g , up to known terms. Under the other assumptions of (c), κ is then identified exactly when each $h \in \mathcal{O}_\kappa$ has an ancestor-or-self in $\mathcal{S}$, by the proof of (c). Minimal sets are then not unique. With the trivial record and a decision at step 0 on every path, storing every step-0 history (the live-run cover) identifies every gate with no snapshot at a running maximum. Re-executing every agreeing step is the price. The claim above is the least set of first-deviation snapshots.

G Proofs: Component Evidence and Composition

This appendix proves Proposition G.1.1, used in Sections 3.5 and 3.6.

G.1 Statement

Section 3.5 states this result in words. Fix a target composition and scheduler, with shared state $S \sim P_S$, component losses L_i with conditional laws $F_i(\cdot \mid s)$, and declared joint loss $L = h(S, L_1, \dots, L_n)$; write $p_i(s) = F_i(B_i \mid s)$. Section G.2 makes these objects precise.

Proposition G.1.1 (Component evidence to joint risk). (i) *If $L_1, \dots, L_n$ are mutually conditionally independent given S (CI), the law of $(S, L_1, \dots, L_n)$ is $P_S(ds) \prod_i F_i(dl_i \mid s)$, and the law of L is its image under h .* (ii) *Let the evidence be P_S and the F_i , as from separate component logs that each carry S , with no restriction on dependence given S (NR). Then $P(L_1 \in B_1, L_2 \in B_2)$ takes exactly the values in $[\int (p_1 + p_2 - 1) dP_S, \int \min(p_1, p_2) dP_S]$.* (iii) *Records linking S and every L_i within each episode identify the law of $(S, L_1, \dots, L_n)$, hence every query of L . Under (NR), separate logs of each (S, L_i) identify that law if and only if, at each state of positive probability, at most one $F_i(\cdot \mid s)$ is not a point mass. Records of S alone identify only P_S in the family of Lemma G.3.2.* (iv) *Standalone runs of component i identify $F_i(\cdot \mid s)$ when the composition's state law is known (Known P_S), the conditional law of L_i given S is the same in standalone runs and in the composition (Inv $_i$), and every composed state occurs in standalone runs (Supp $_i$). In a family that does not tie the composed component to its standalone run, the standalone law of (S, L_i) places no restriction on F_i . Without (Supp $_i$), let component i read only S and private randomness, and let U_i hold the composed states never seen in isolation. For $\emptyset \neq B \subsetneq \mathbb{L}_i$, the identified set of $P(L_i \in B)$ is then an interval of width $P_S(U_i)$, and other restrictions on component i can narrow it.*

Parts (i) to (iii) restate conditional independence, Fréchet bounds given S (Fréchet, 1951; Nelsen, 2006), and linked-record identification; part (iv) gives transportability with overlapping support (Pearl and Bareinboim, 2011).

G.2 Finite Setting and Composition Interfaces

This setting supplies the objects and assumptions for Proposition G.1.1 (Section 3.5).

A composition declares interfaces, a scheduler, compatible kernels, and joint loss map $h : \mathcal{S} \times \mathbb{L}_{1:n} \rightarrow \mathbb{L} \subset \mathbb{R}^m$. This defines joint specification $\mathfrak{R}_{\text{joint}} = (E_{\text{joint}}, \ell_{\text{joint}}, K_{\text{joint}}, C_{\text{joint}})$ with law P

on finite trajectory space Ω .

Shared state is a declared map $S : \Omega \rightarrow \mathcal{S}$, with finite $\mathcal{S}$. Component $i \in \{1, \dots, n\}$ has loss map $L_i : \Omega \rightarrow \mathbb{L}_i$ with finite $\mathbb{L}_i$. Write $\mathbf{L} = (L_1, \dots, L_n) \in \mathbb{L}_{1:n}$, consequence map $\ell_{\text{joint}} = h \circ (S, \mathbf{L})$, and full-episode loss $L = h(S, \mathbf{L})$. Let $\Lambda_P(s, \mathbf{l}) = P(S = s, \mathbf{L} = \mathbf{l})$ be the law of $(S, \mathbf{L})$, with state marginal $P_S(s) = \sum_{\mathbf{l}} \Lambda_P(s, \mathbf{l})$, support $\mathcal{S}_+ = \{s : P_S(s) > 0\}$, and conditional laws $F_i(l \mid s) = P(L_i = l \mid S = s)$ for $s \in \mathcal{S}_+$. For events $B_i \subseteq \mathbb{L}_i$, write $E_i = \{L_i \in B_i\}$ and $p_i(s) = F_i(B_i \mid s)$. The product law is

$$\Lambda^\otimes(s, \mathbf{l}) = P_S(s) \prod_{i=1}^n F_i(l_i \mid s). \quad (18)$$

For component i , standalone deployment is a mechanism with law P_i^{sa} yielding evidence $Q^{(i)} = \mathcal{L}_{P_i^{\text{sa}}}(S, L_i)$, state marginal $Q_S^{(i)}$, and support $\mathcal{S}_i^{\text{sa}} = \{s : Q_S^{(i)}(s) > 0\}$. The model family $\mathcal{M}_0$ contains finite composition mechanisms under the scheduler, with true mechanism $M^* \in \mathcal{M}_0$ having $P_{M^*} = P$ and $\Lambda_{M^*} = \Lambda_P$. Identified sets are $\Theta_e(\theta) = \{\theta(M) : M \in \mathcal{M}_0, e(M) = e(M^*)\}$. Evidence maps are linked records $e_{\text{link}}(M) = \mathcal{L}_{P_M}(S, \mathbf{L})$, paired records $e_{\text{pair}}(M) = (\mathcal{L}_{P_M}(S, L_1), \dots, \mathcal{L}_{P_M}(S, L_n))$, and state records $e_S(M) = \mathcal{L}_{P_M}(S)$.

Definition G.2.1 (Coupling set). $\Gamma = \Gamma(P_S; F_1, \dots, F_n)$ is the set of probability laws Λ on $\mathcal{S} \times \mathbb{L}_{1:n}$ such that, for every $s \in \mathcal{S}$, every i , and every $l \in \mathbb{L}_i$,

$$\sum_{\mathbf{l}: l_i=l} \Lambda(s, \mathbf{l}) = P_S(s) F_i(l \mid s).$$

Summing over l shows every $\Lambda \in \Gamma$ has state marginal P_S , and for each $s \in \mathcal{S}_+$, $\pi_s = \Lambda(s, \cdot) / P_S(s)$ couples $F_1(\cdot \mid s), \dots, F_n(\cdot \mid s)$; thus $\Lambda^\otimes, \Lambda_P \in \Gamma$.

Definition G.2.2 (Assumptions). (CI) $L_1, \dots, L_n$ are mutually conditionally independent given S under P : $P(\mathbf{L} = \mathbf{l} \mid S = s) = \prod_i F_i(l_i \mid s)$ for every $s \in \mathcal{S}_+$ and every $\mathbf{l}$.

(NR) No dependence restriction: $\Gamma(P_S; F_1, \dots, F_n) \subseteq \{\Lambda_M : M \in \mathcal{M}_0\}$.

(Known P_S) The shared-state law of the target composition is available, from logged S in that composition or from its declared scheduler and upstream kernels.

(Inv $_i$) There is a kernel G_i from $\mathcal{S}$ to $\mathbb{L}_i$ with $F_i(\cdot \mid s) = G_i(\cdot \mid s)$ for every $s \in \mathcal{S}_+$ and

$Q^{(i)}(L_i \in \cdot \mid S = s) = G_i(\cdot \mid s)$ for every $s \in \mathcal{S}_i^{\text{sa}}$.
 $(\text{Supp}_i) \mathcal{S}_+ \subseteq \mathcal{S}_i^{\text{sa}}$.

G.3 Lemmas

These lemmas support the proof of Proposition G.1.1 (Section 3.5).

Lemma G.3.1 (Paired evidence fixes exactly the coupling set). *For $M \in \mathcal{M}_0$, $e_{\text{pair}}(M) = e_{\text{pair}}(M^*)$ if and only if $\Lambda_M \in \Gamma(P_S; F_1, \dots, F_n)$, where P_S and the F_i are those of M^* .*

Proof. The marginal conditions $\mathcal{L}_{P_M}(S, L_i)(s, l) = \sum_{\mathbf{l}: l_i=l} \Lambda_M(s, \mathbf{l}) = P_S(s)F_i(l \mid s)$ for all i, s, l define Γ . $\square$

Lemma G.3.2 (Realization by an unlogged shared input). *Let $\mathcal{M}_0^W$ be the family of mechanisms in which a shared-state kernel draws S , an unlogged exogenous input V on a finite set is drawn from a kernel given S , and each component i , in the scheduled order, outputs $L_i = k_i(S, V)$ for a deterministic map k_i . Every law Λ on $\mathcal{S} \times \mathbb{L}_{1:n}$ is Λ_M for some $M \in \mathcal{M}_0^W$. In particular $\mathcal{M}_0^W$ satisfies (NR) at every $(P_S, F_1, \dots, F_n)$.*

Proof. Draw $S \sim \Lambda_S$, draw $V \sim \Lambda(\cdot \mid S)$, and set $k_i(s, \mathbf{v}) = v_i$. $\square$

Lemma G.3.3 (When the coupling set is a point). Γ is a singleton if and only if, for every $s \in \mathcal{S}_+$, at most one of $F_1(\cdot \mid s), \dots, F_n(\cdot \mid s)$ is not a point mass.

Proof. If at most one margin is non-degenerate at each $s \in \mathcal{S}_+$, the marginal constraints uniquely fix $\Lambda(s, \cdot)$. Otherwise, choose two distinct positive-mass values in each of two non-degenerate margins at some $s \in \mathcal{S}_+$, and fix positive-mass values in all other margins. The four resulting cells have positive mass under $\prod_i F_i(\cdot \mid s)$. Add $\epsilon > 0$ to the two diagonal cells and subtract it from the other two, with ϵ smaller than their masses. This preserves every margin and non-negativity; multiplying by $P_S(s)$ and leaving other states unchanged gives a second member of Γ . $\square$

G.4 Proofs of Proposition G.1.1 Parts (i) to (iv)

Proof of Part (i). For $s \in \mathcal{S}_+$ and $\mathbf{l}$, conditional probability and (CI) give $\Lambda_P(s, \mathbf{l}) = P_S(s) \prod_i F_i(l_i \mid s) = \Lambda^\otimes(s, \mathbf{l})$, vanishing outside $\mathcal{S}_+$. Because $L = h(S, \mathbf{L})$, $\mathcal{L}_P(L) = h_\# \Lambda^\otimes$, so every query of L is a function of $(P_S, F_1, \dots, F_n)$.

Paired records cannot test (CI) because $\Lambda^\otimes \in \Gamma$ and part (iii)(b) shows they exclude no element of Γ under (NR).

Remark G.4.1 (Sub-parts of part (ii)). Let $\beta^-(s) = (p_1(s) + p_2(s) - 1)_+$ and $\beta^+(s) = \min(p_1(s), p_2(s))$ for $s \in \mathcal{S}_+$. For a function β on $\mathcal{S}_+$, write $\bar{\beta} = \sum_s P_S(s)\beta(s)$; in particular $\bar{\beta}^- = \sum_s P_S(s)\beta^-(s)$ and $\bar{\beta}^+ = \sum_s P_S(s)\beta^+(s)$. (a) Every $\Lambda \in \Gamma$, in particular Λ_P , satisfies $\beta^-(s) \leq \Lambda(E_1 \cap E_2 \mid S = s) \leq \beta^+(s)$ for $s \in \mathcal{S}_+$, and $\bar{\beta}^- \leq \Lambda(E_1 \cap E_2) \leq \bar{\beta}^+$. (b) Under (NR), $\Theta_{e_{\text{pair}}}(P(E_1 \cap E_2)) = [\bar{\beta}^-, \bar{\beta}^+]$. (c) $(P(E_1) + P(E_2) - 1)_+ \leq \bar{\beta}^-$ and $\bar{\beta}^+ \leq \min(P(E_1), P(E_2))$. (d) Under (NR), the identified set for the law of L under paired records is $\mathcal{H} = \{h_\# \Lambda : \Lambda \in \Gamma\}$, and that of a query (ϕ, ρ) is $\{\rho(\phi_\# \nu) : \nu \in \mathcal{H}\}$. If $n = 2$ and $h(s, l_1, l_2) = \bar{h}(\mathbf{1}_{B_1}(l_1), \mathbf{1}_{B_2}(l_2))$ does not depend on s , each element $h_\# \Lambda$ of $\mathcal{H}$ depends on Λ only through $\bar{\beta} = \Lambda(E_1 \cap E_2)$, which ranges over $[\bar{\beta}^-, \bar{\beta}^+]$.

Proof of Part (ii). (a) At each $s \in \mathcal{S}_+$, apply the Fréchet bounds (Fréchet, 1951), in Nelsen (2006, Theorem 2.2.3 and Eq. (2.5.1)), to $1 - \mathbf{1}_{B_1}(L_1)$ and $1 - \mathbf{1}_{B_2}(L_2)$ conditionally on $S = s$, evaluating their joint CDF at $(0, 0)$. These finite conditional laws have CDF values $p_1(s), p_2(s)$ there, giving $\beta^\pm(s)$; averaging gives $\bar{\beta}^\pm$. (c) follows by summing $(x)_+ \geq x, 0$ and $\min(x, y) \leq x, y$ against P_S . (b) The bounds in Nelsen's Eq. (2.5.1) are themselves joint CDFs, including for discrete margins. Mix the two endpoint indicator couplings with a common weight $\lambda \in [0, 1]$ at every s . Lift each positive indicator cell to the full losses by taking the product of the $F_i(\cdot \mid s)$ conditioned on that cell's indicators; positive cell mass ensures these conditional laws exist. Give components $i \geq 3$ their independent margins. This yields a member of Γ with intersection mass $(1 - \lambda)\bar{\beta}^- + \lambda\bar{\beta}^+$. By (NR) and Lemma G.3.1, each such coupling has a compatible mechanism, proving sharpness and every intermediate value. (d) By Lemma G.3.1 and (NR), the consistent $(S, \mathbf{L})$ -laws form exactly Γ , and $\mathcal{L}(L) = h_\# \Lambda$. Let $n = 2$ and let $\bar{h}$ be an indicator map that does not depend on s . The law of $(\mathbf{1}_{B_1}(L_1), \mathbf{1}_{B_2}(L_2))$ under $\Lambda \in \Gamma$ has the fixed margins $P(E_1)$ and $P(E_2)$ and joint cell $\bar{\beta} = \Lambda(E_1 \cap E_2)$. Hence $h_\# \Lambda$ depends on Λ only through $\bar{\beta}$, which ranges over $[\bar{\beta}^-, \bar{\beta}^+]$ by (b).

Proof of Part (iii). (a) Linked records $\log(S, \mathbf{L})$ jointly, so in every family $e_{\text{link}}(M) = e_{\text{link}}(M^*)$ implies $\Lambda_M = \Lambda_P$, which identifies every query of L through $\mathcal{L}_P(L) = h_{\#}\Lambda_P$. (b) By Lemma G.3.1, $\Theta_{e_{\text{pair}}}(\Lambda) = \{\Lambda_M : M \in \mathcal{M}_0\} \cap \Gamma$, which equals Γ under (NR). (c) $e_S(M)$ is the state marginal of each component of $e_{\text{pair}}(M)$, so $\Theta_{e_{\text{pair}}}(\Lambda) \subseteq \Theta_{e_S}(\Lambda)$. The family $\mathcal{M}_0^W$ of Lemma G.3.2 realizes every law with state marginal P_S and satisfies (NR). In it, $e_S(M) = e_S(M^*)$ holds exactly when Λ_M has state marginal P_S , and two elements $\mathbf{l} \neq \mathbf{l}'$ of $\mathbb{L}_{1:n}$ give the distinct members $\sum_s P_S(s)\delta_{(s,\mathbf{l})}$ and $\sum_s P_S(s)\delta_{(s,\mathbf{l}')}$, so records of S alone identify only P_S . (d) Lemma G.3.3 combined with (b) shows that, under (NR), paired records identify Λ_P exactly when, for every $s \in \mathcal{S}_+$, at most one $F_i(\cdot | s)$ is not a point mass.

Remark G.4.2 (Sub-parts of part (iv)). Assume (Known P_S). (a) Under (Inv_i) and (Supp_i) , $F_i(\cdot | s) = Q^{(i)}(L_i \in \cdot | S = s)$ for every $s \in \mathcal{S}_+$. If this holds for every i , then under (CI) the law of L is $h_{\#}\Lambda^{\otimes}$ computed from standalone evidence and P_S , and without (CI) the bounds of Remark G.4.1(a) hold with these F_i . (b) Let $U_i = \mathcal{S}_+ \setminus \mathcal{S}_i^{\text{sa}}$, and let $\mathcal{M}_0^{\text{tr},i}$ be the family of pairs of a composition and a standalone deployment in which component i is one kernel $G_i(\cdot | S)$, reading only S and private randomness in both, with G_i arbitrary, composition state law P_S , and standalone state law $Q_S^{(i)}$. In this family (Inv_i) holds, and for every B with $\emptyset \neq B \subsetneq \mathbb{L}_i$, the identified set of $P(L_i \in B)$ given $Q^{(i)}$ and P_S is

$$c_B = \sum_{s \in \mathcal{S}_+ \setminus U_i} P_S(s) Q^{(i)}(L_i \in B | S = s),$$

It is a singleton if and only if (Supp_i) holds. (c) Without (Inv_i) , $Q^{(i)}$ places no restriction on F_i : for every family of laws $(F_i'(\cdot | s))_{s \in \mathcal{S}_+}$ on $\mathbb{L}_i$, some composition with state law P_S has these conditional laws while its standalone deployment has evidence $Q^{(i)}$.

Proof of Part (iv). (a) Under (Supp_i) , each $s \in \mathcal{S}_+$ lies in $\mathcal{S}_i^{\text{sa}}$, so (Inv_i) gives $F_i(\cdot | s) = G_i(\cdot | s) = Q^{(i)}(L_i \in \cdot | S = s)$. With (Known P_S), $\Lambda^{\otimes}$ is computable, part (i) gives $\mathcal{L}_P(L) = h_{\#}\Lambda^{\otimes}$ under (CI), and Remark G.4.1(a) applies without (CI) because it uses only membership in Γ . (b) Component i reads only S and private randomness in both deployments, so (Inv_i) holds in $\mathcal{M}_0^{\text{tr},i}$. For

$s \in \mathcal{S}_+ \setminus U_i$, (Inv_i) fixes $F_i(B | s) = Q^{(i)}(L_i \in B | S = s)$, contributing c_B , while $G_i(B | s)$ is unconstrained in $[0, 1]$ on U_i , so $P(L_i \in B) = c_B + \sum_{s \in U_i} P_S(s) G_i(B | s) \in [c_B, c_B + P_S(U_i)]$. For attainment, fix $x_{\text{in}} \in B$ and $x_{\text{out}} \notin B$, set $G_i(\cdot | s) = Q^{(i)}(L_i \in \cdot | S = s)$ on $\mathcal{S}_i^{\text{sa}}$ and $\lambda\delta_{x_{\text{in}}} + (1 - \lambda)\delta_{x_{\text{out}}}$ on U_i ; the standalone deployment that draws S from $Q_S^{(i)}$ then has record law $Q^{(i)}$, and the composition gives $c_B + \lambda P_S(U_i)$ for $\lambda \in [0, 1]$. Since $P_S > 0$ on U_i , the interval is a singleton if and only if $U_i = \emptyset$, which is (Supp_i) . (c) A standalone deployment that draws S from $Q_S^{(i)}$ and L_i from $Q^{(i)}(L_i \in \cdot | S)$, paired with a composition that produces S with law P_S and draws L_i from $F_i'(\cdot | S)$, has evidence $Q^{(i)}$ and conditional laws F_i' . In a deployment, different kernels at the same logged S arise when an unlogged input differs between the two deployments.

Remark G.4.3 (Oversight cues). Let the composition be deployment and the standalone deployment be testing, with one component, the agent, and the cue Z as S . When Z is recorded and the agent's kernel reads only Z and private randomness, Remark G.4.2(b) gives the interval $[c_B, c_B + P_{\text{dep}}(U)]$ for a deployment event, where U holds the cue values never seen in testing.

H Verification Index

Scripts supporting Sections 2 and 3 evaluate finite instances using exact rational arithmetic (fractions.Fraction); seeded Monte Carlo cases, logarithms, and dynamic programs use floating point as disclosed in their headers. These calculations verify finite instances and counterexamples, whereas Appendices A to G prove the general theorems. The companion file checks/README.md provides the full script-to-claim map, execution commands, and worked examples.

I AI Use Statement

This disclosure supplements the Ethical Considerations.

The authors used AI coding and writing assistants to support implementation, analysis tooling, proof checking, figure preparation, and manuscript drafting and revision. The reported empirical results are computed by evaluation code over logged runs and saved experimental artifacts. The authors take full responsibility for the methods, interpretation, and final content of the paper.