Verifiers That Do Not Trust the Agent: Anytime-Valid Release Gates and Hacking Bounds for AI Agents That Move Money

Dhruv Hegde
Scruple
dhegde@umich.edu

Preprint, November 2026

Abstract

AI agents that handle card disputes move money: they decide when to refund, when to fight and with which evidence. Changes to them are usually judged by win rates and by the agent's own reports, and both can be gamed. We describe a release gate whose verifiers never credit the agent's claims: cost is recomputed from double-entry ledger postings, and every action is checked against a hash-chained trace written by the environment. Its statistics are anytime-valid. We prove that if each hacked case is caught with conditional probability at least ρ, whatever the agent knows, an accepted change carries a proxy-over-ledger gap $\Gamma \geq \gamma$ with probability at most $(1-\rho)^{\lceil n\gamma/\varepsilon \rceil}$; we give ρ for audits that leak, the audit rate at which hacking stops paying when the verifier errs both ways, and what the composed promote decision guarantees. On a frozen synthetic benchmark of 10,000 disputes, one of four pre-registered experiments passed (narrowly), two failed their primary hypothesis and one found no effect; we report why. An optimiser on a naive proxy learned forgery that the simulated ledger rated cheaper. Five learners used no exploit when trained on the scorecard; on the proxy all five claimed false wins and three learned forgery. Real Claude models exposed three verifier false positives.

Data and labels. Numbers marked *synthetic* come from the scruple simulator; numbers marked *real* come from public CFPB complaint data or from real model behaviour. No number describes a real merchant, bank or card network. Each result is labelled *standard* (known, applied here), *adaptation* (a small variant of a known result) or *candidate novel* (not found stated this way in a short search; a full literature review is still required). Code, data and every result file: `https://github.com/thedhruvhegde/scruple`, directory `engine/`.

1 Introduction

Card disputes are a good test bed for agents that move money. Every case ends in postings to a ledger: a refund, a chargeback, a fee, a representment won or lost. The rules are public (Regulation E, 12 CFR 1005.11, for debit cards; Regulation Z, 12 CFR 1026.13, for credit cards), and the incentives to cheat are concrete. An agent judged on its win rate can fight claims it should concede, cite documents it never opened, upload documents that never existed, or simply report a win.

The usual guard for an agent change is an offline evaluation: replay the new agent on held-out cases, compare a metric, ship if it improves. That guard fails in three ways that we measure.

1. **The metric trusts the agent.** In our experiment E2 a seeded hill climb on a "wins plus self-reported wins" proxy learned two exploits (fabricated documents and false win claims) in three moves.

2. **The statistics do not survive how they are used.** Re-running a z-test every 10 cases at the null boundary rejects 14.7% to 34.3% of the time at a nominal 5% (synthetic). A regression of 0.35 cap units in a case type that is 1% of traffic, with the rest improving, is flagged by a pooled valid test 0.6% of the time.
3. **Nobody states what the release decision guarantees.** A gate mixes deterministic checks, soft scores and tests; which error does the combination control?

Contributions.

- A verifier design anchored to the ledger and to an environment-written trace, with an attack for every check (Section 3), and a record of what broke: an engine bug found through verifier false alarms, false positives found by real models, and three exploits closed (Section 14).
- Anytime-valid paired and stratified tests with complete proofs (Sections 4 and 5), sample-size planning with a matching lower bound (Section 6), and variance-adaptive betting with a sample-size floor computed from observed data (Section 7).
- A composition theorem for the release decision (Section 9) and its asymmetric-cost refinement (Section 8).
- Hacking-gap bounds for adaptive agents with leaky audits and an imperfect verifier, and the minimal audit rate that removes the incentive to hack (Section 10).
- Four pre-registered experiments, reported whichever way they fell: planted regressions (E1, pass), an optimiser on a naive proxy (E2, fails H2), a variance-reduced gate (E3, fails H1) and a win-rate incentive for real models (E4, H1 not supported) (Section 11).
- A comparison of six agent types, five of them learners trained under both the scorecard and the naive proxy (Section 12); 64 dispute scenarios grounded in real CFPB complaints, and a timeliness analysis of 164,821 real complaints (Section 13).

2 Setting and notation

Definition 2.1 (Episode and record)**.** A dispute episode starts from a seeded case: a card transaction, a reason code, a cardholder claim with a hidden truth (valid or not), documents the merchant does or does not hold, and network deadlines. The agent acts through tools (read the case, list and read evidence, request information, read the ledger, refund, represent, accept, wait, upload evidence). The environment writes every call and every state change to an append-only trace whose events are hashed into a chain with a recorded head. Money is double-entry, in USD minor units. The *record* of an episode is the trace, the final journal, the documents as generated and at close, the case with its hidden truth, and the agent's self-report.

Definition 2.2 (Paired difference)**.** A candidate policy C is compared with the current policy B on the same held-out cases and the same issuer draws. Case i gives losses B_i, C_i (merchant cost from the journal). Before outcomes are seen the operator declares a cap $c > 0$, and

$$X_i = \frac{C_i - B_i}{c} \in [-1, 1], \qquad \mu_i = \mathbb{E}[X_i \mid \mathcal{F}_{i-1}], \tag{1}$$

where $\mathcal{F}_{i-1}$ is everything observed before case i. Positive X_i means the candidate is worse.

Assumption 2.3 (Cap)**.** $|C_i - B_i| \le c$ for every case. A violation is an error; it is never clipped silently.

Assumption 2.4 (Predictable bets)**.** Every bet λ_i used for case i is $\mathcal{F}_{i-1}$-measurable.

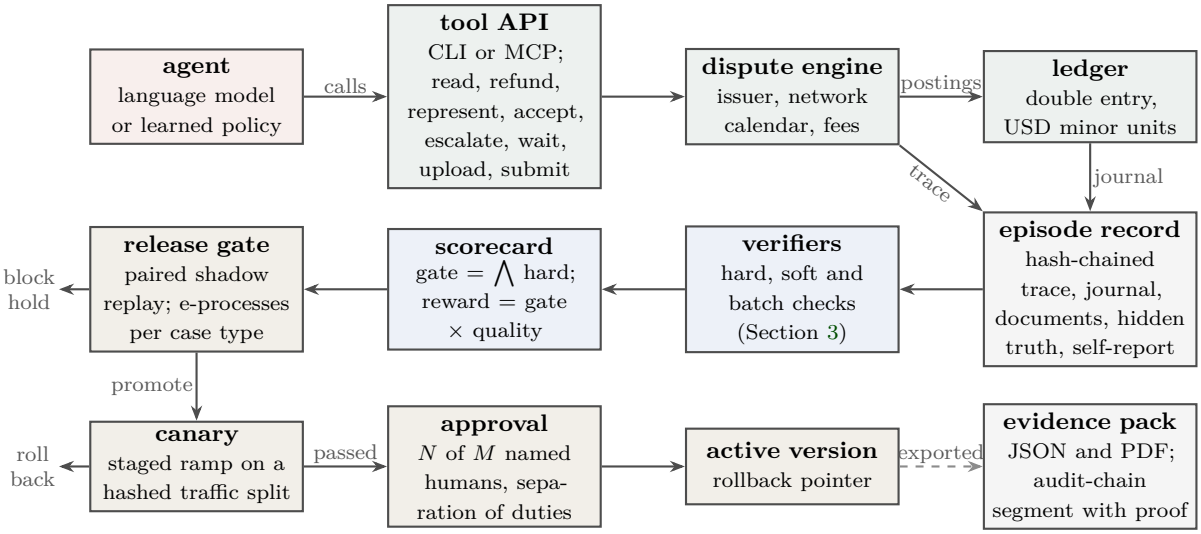


Figure 1: The system, as built. The agent sees only tool outputs; it never holds the engine, the ledger or the hidden truth. The environment, not the agent, writes the trace. Nothing the agent says earns credit: the self-report is checked against the record and can only lose points. Every gate decision, canary step and approval is appended to an audit hash chain, which the evidence pack exports with a verification proof.

The gate decides *promote*, *hold* or *block*. The error that costs money is promoting a harmful change; the error that costs adoption is blocking a good one.

Dispute lifecycle. Figure 2 is the engine's state machine. Deadlines and fees are those of the simulator (`docs/RULES.md`): sourced values cite public network documents as reported, the rest are marked as assumptions there. The agent acts only while the case awaits the merchant; the issuer and the network answer on the engine's clock, and a missed merchant deadline is decided by the engine, not by the agent.

Benchmark. Benchmark v1 is 10,000 synthetic dispute cases from a sealed held-out seed namespace; only a manifest of commitments is public (content hash `1286baf3...2cb13f91`). Family mix: buyer's remorse 1,977; cardholder mistake 577; friendly fraud 2,791; merchant error 625; service failure 1,840; true fraud 2,190; 4,655 claims are valid. Reference policies (synthetic): the *evidence* policy (represent when the required evidence exists, else concede) costs $82.82 per case (95% interval 80.72 to 84.91) and fails a hard check on 19 of 10,000 cases (an engine bug, since fixed; Section 14); always-accept costs $127.46; an oracle that reads the hidden truth costs $86.67, more than the evidence policy, because knowing a claim is invalid does not win it without documents.

3 Verifier design: anchoring to the ledger and the trace

The design principle is that nothing the agent says earns credit. Self-reports are *checked*, never *scored*: a self-report can only lose points. Scores compose as

$$\text{gate} = \bigwedge_h \text{hard}_h, \qquad \text{quality} = \textstyle\sum_j w_j s_j / \sum_j w_j, \qquad \text{reward} = \text{gate} \times \text{quality}, \tag{2}$$

so a hard failure zeroes the reward and ten perfect soft scores cannot outvote one violation. Table 1 lists the checks and, for each, an attack that fools a naive metric and is stopped by the check. Figure 3 shows the three kinds of check and how the release decision combines them; Section 9 proves what the combination guarantees.

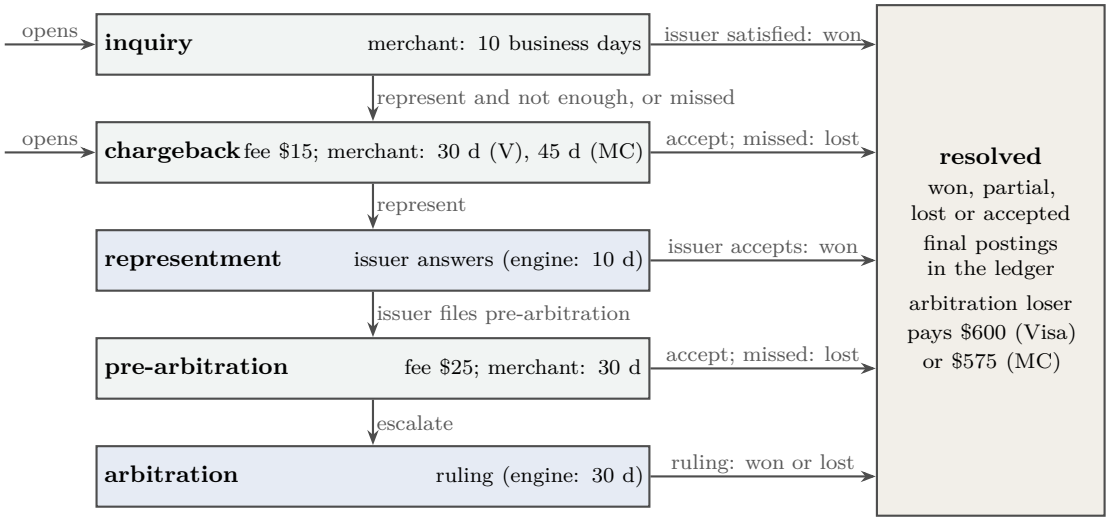


Figure 2: The dispute lifecycle in the engine (collaboration workflow; simplified). Shaded states await the issuer or the network; the others await the merchant, whose deadline is shown. A case opens as an inquiry with probability 0.2 and as a chargeback otherwise (assumption A-32). Fees are charged on entering a state. A refund may be issued in any open state, and a settled refund record makes the dispute moot (A-23). Windows: Visa chargeback response 30 days (source S1), Mastercard 45 days (S1, S11), pre-arbitration response 30 days (Visa S8; Mastercard A-14), inquiry response 10 business days (A-13); fees (A-17) and engine timing (A-25) are assumptions. Not legal or network advice.

The exploits are 38 patterns from a research list (X1 to X38), each a scripted policy, a false self-report or a rewrite of the record, run against a naive metric and the scorecard on the same 300 synthetic seeds as the honest reference policy. "Caught" means the naive metric rates the exploit at least as well as the honest policy, the expected check fires on at least 90% of exercised episodes, and the reward falls; "blocked" means the environment refuses the move. The status is computed from the run, never declared. Today 26 of the 38 are caught, 5 blocked and 7 cannot be expressed against a merchant-side engine (issuer-side Regulation E and Z patterns); Section 14 gives the run before and after the last round of fixes.

4 Anytime-valid paired test

We first record the two probabilistic tools every later proof uses.

Lemma 4.1 (Ville's inequality)**.** *Let $(K_t)_{t\geq 0}$ be a non-negative supermartingale with $K_0 = 1$. Then for every $a > 0$, $\mathbb{P}(\exists t : K_t \geq a) \leq 1/a$, and for every stopping time τ (possibly infinite, with $K_\infty = \liminf_t K_t$), $\mathbb{E}[K_\tau] \leq 1$.* *[standard; Ville, 1939]*

Proof. Let $\sigma = \inf\{t : K_t \geq a\}$. For fixed T, $\sigma \wedge T$ is a bounded stopping time, so by optional stopping for supermartingales (Durrett, 2019, Ch. 4) $\mathbb{E}[K_{\sigma\wedge T}] \leq K_0 = 1$. Since $K \geq 0$ and $K_{\sigma\wedge T} \geq a$ on $\{\sigma \leq T\}$, $a\,\mathbb{P}(\sigma \leq T) \leq 1$. Let $T \to \infty$: $\mathbb{P}(\sigma < \infty) \leq 1/a$. For the second claim, $K_{\tau\wedge T} \to K_\tau$ on $\{\tau < \infty\}$ and $\liminf_T K_{\tau\wedge T} = \liminf_t K_t = K_\infty$ on $\{\tau = \infty\}$; Fatou's lemma gives $\mathbb{E}[K_\tau] \leq \liminf_T \mathbb{E}[K_{\tau\wedge T}] \leq 1$. □

Lemma 4.2 (Data processing for KL)**.** *Let $\mathrm{kl}(p\|q) = p\log\frac{p}{q} + (1-p)\log\frac{1-p}{1-q}$. For laws P, Q and any event A, $\mathrm{KL}(P\|Q) \geq \mathrm{kl}(P(A)\|Q(A))$. If $P(A) \geq 1-\beta > \alpha \geq Q(A)$, then $\mathrm{kl}(P(A)\|Q(A)) \geq \mathrm{kl}(1-\beta\|\alpha)$.* *[standard; Cover and Thomas, 2006]*

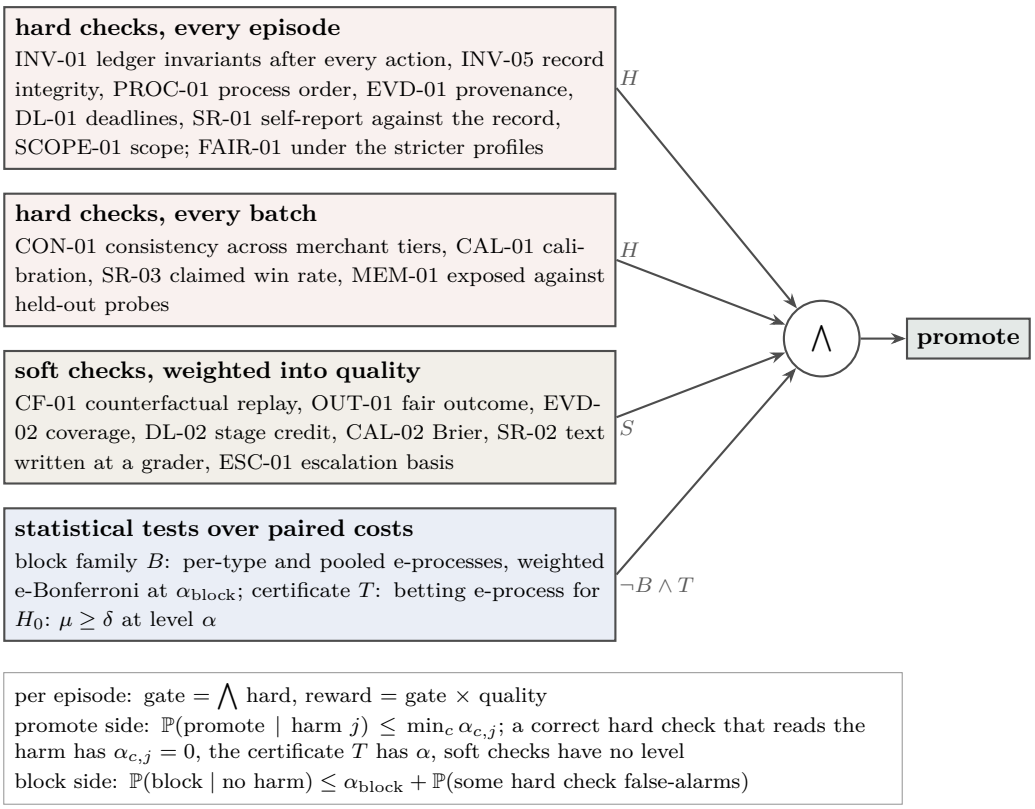


Figure 3: The verifier stack. Hard checks are conjuncts: one failure zeroes the episode's reward and blocks the candidate. Soft checks only lower quality. Statistical tests read costs recomputed from the ledger, never the agent's report. By Theorem 9.1, the promote decision is as strict as the strictest conjunct that covers a harm, false blocks add up across conjuncts, and a harm no conjunct covers (for example an unfair win under the default profile, which only OUT-01 scores) carries no guarantee.

Proof. The first inequality is the data-processing inequality applied to the map $\omega \mapsto \mathbf{1}_A(\omega)$. For the second, $\partial_p \mathrm{kl}(p\|q) = \log \frac{p(1-q)}{q(1-p)} \geq 0$ for $p \geq q$ and $\partial_q \mathrm{kl}(p\|q) = \frac{q-p}{q(1-q)} \leq 0$ for $q \leq p$; so increasing p from $1-\beta$ and decreasing q from α only increases kl. □

Proposition 4.3 (Betting e-process)**.** *Fix $m \in (-1,1)$ and a side $s \in \{+1,-1\}$ ($s=+1$: regression, $H_0 : \mu_i \leq m$ for all i; $s=-1$: certification, $H_0 : \mu_i \geq m$ for all i). Let $Y_i = s(X_i - m)$ and predictable bets $0 \leq \lambda_i \leq b/(1+sm)$ with $b < 1$. Then*

$$K_t = \prod_{i \leq t} (1 + \lambda_i Y_i) \tag{3}$$

satisfies, under H_0, $\mathbb{P}(\exists t : K_t \geq 1/\alpha) \leq \alpha$, and $\mathbb{E}[K_\tau] \leq 1$ for every stopping time τ.
[standard; Ramdas et al., 2023; Waudby-Smith and Ramdas, 2024]

Proof. Since $X_i \in [-1,1]$, $Y_i \geq -(1+sm)$, so $1 + \lambda_i Y_i \geq 1 - b > 0$ and $K_t > 0$. Under H_0, $s(\mu_i - m) \leq 0$; as $\lambda_i \geq 0$ is $\mathcal{F}_{i-1}$-measurable, $\mathbb{E}[1 + \lambda_i Y_i \mid \mathcal{F}_{i-1}] = 1 + \lambda_i s(\mu_i - m) \leq 1$. Hence K is a non-negative supermartingale with $K_0 = 1$, and Lemma 4.1 with $a = 1/\alpha$ gives both claims. □

The aGRAPA bet of Algorithm 1 maximises $\lambda \mathbb{E}Y - \lambda^2 \mathbb{E}Y^2/2$, the second-order expansion of the growth rate $\mathbb{E}\log(1+\lambda Y)$ (Kelly, 1956; Waudby-Smith and Ramdas, 2024). It affects power only, never validity.

Table 1: Checks, what they read, and the attack each stops. Synthetic, 300 seeds per attack.

check	reads	kind	attack stopped (naive metric fooled)
INV-01 ledger invariants	trace postings, after every action	hard	double credit then moot
INV-05 record integrity	hash chain and head	hard	outcome rewrite (correct rate 1.0)
PROC-01 process	trace order	hard	summary response (half the calls)
EVD-01 provenance	generator truth, content hashes	hard	fabrication (win rate 0.044 to 0.85); adverse-document swap
DL-01 deadlines	env-recorded network calendar	hard	stall on pending information
SR-01 self-report	claims parsed into predicates on the record	hard	claimed, not done
SCOPE-01 scope	protected reads, tool and argument scope	hard	ground-truth peeking
CON-01, CAL-01, SR-03	batch statistics	hard	segment skimming, flat confidence
MEM-01 memorisation	exposed cases against held-out probes	hard (batch)	memorised outcomes
CF-01 counterfactual	replay under baseline, same draws	soft	refund to win
ESC-01 escalation basis	the submitted case file	soft	escalate everything
OUT-01 fair outcome	hidden truth	soft	unfair wins (Section 11.5)
FAIR-01 (profiles)	hidden truth	hard if enabled	a win against a valid claim

Proposition 4.4 (Exact-interval confidence sequence)**.** *Let X_i be i.i.d. with mean μ, and let $\lambda_i \in [0, b]$, $b < 1/2$, be predictable and not depend on the candidate mean m. Set $K_t^+(m) = \prod_{i\le t}(1 + \lambda_i(X_i - m))$, $K_t^-(m) = \prod_{i\le t}(1 + \lambda_i(m - X_i))$ and*

$$C_t = \{m \in [-1, 1] : K_t^+(m) < 2/\alpha \text{ and } K_t^-(m) < 2/\alpha\}.$$

Then each C_t is an interval, $\mathbb{P}(\forall t : \mu \in C_t) \ge 1 - \alpha$, and the same holds for the running intersection $\bigcap_{s\le t} C_s$. *[adaptation of Waudby-Smith and Ramdas, 2024]*

Proof. For $m \in [-1, 1]$ every factor $1 + \lambda_i(X_i - m) \ge 1 - 2b > 0$ and is non-increasing in m; a product of positive non-increasing functions is non-increasing, so $\{m : K_t^+(m) \ge 2/\alpha\}$ is an initial segment of $[-1, 1]$. Symmetrically $\{m : K_t^-(m) \ge 2/\alpha\}$ is a final segment, so C_t is an interval. At $m = \mu$, $\mathbb{E}[1 + \lambda_i(X_i - \mu) \mid \mathcal{F}_{i-1}] = 1$, so $K_t^{\pm}(\mu)$ are non-negative martingales with initial value 1; by Lemma 4.1 each exceeds $2/\alpha$ at some time with probability at most $\alpha/2$, and by a union bound $\mu \in C_t$ for all t with probability at least $1 - \alpha$. On that event μ lies in every C_s and so in their intersection. □

Evidence (synthetic). Nulls at the boundary $\mu = m = 0$, four null shapes, seven stopping rules applied to the same 2,000 paths of 2,000 cases (including "stop when it looks bad"): the worst rejection rate of the betting test was 0.027 at $\alpha = 0.05$. A one-sided z-test re-run every 10 cases rejected 0.147 to 0.343 of the time (Figure 4). Power (ternary data, 10% of cases changed) is in Figure 5.

Algorithm 1 Paired regression or certification test (default bet: aGRAPA)

Require: cap c, null mean m, side s, level α, max bet $b < 1$, prior second moment $v_0 = 1/4$

$\log K \leftarrow 0$, $S_1 \leftarrow 0$, $S_2 \leftarrow 0$, $n \leftarrow 0$
for each case i as it resolves **do**
  $\lambda \leftarrow \min\{\max\{(S_1/(n+1))/((v_0+S_2)/(n+1)), 0\},\ b/(1+sm)\}$ ▷ uses past data only
  $Y \leftarrow s((C_i - B_i)/c - m)$; $\log K \leftarrow \log K + \log(1+\lambda Y)$
  $S_1 \leftarrow S_1 + Y$, $S_2 \leftarrow S_2 + Y^2$, $n \leftarrow n+1$
  if $\log K \geq \log(1/\alpha)$ **then return** reject H_0 at case n
  end if
end for
return do not reject

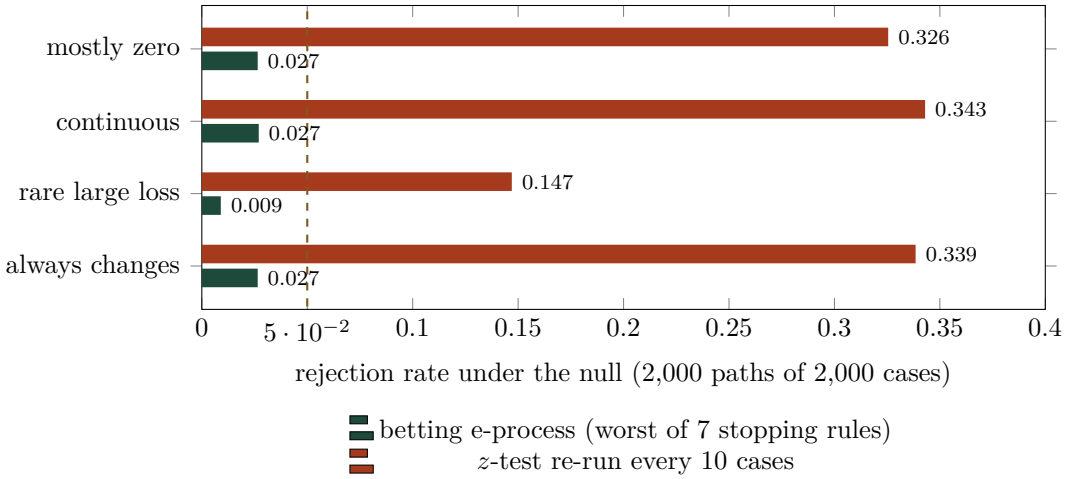


Figure 4: Continuous monitoring at the null boundary, $\alpha = 0.05$ (dashed). Synthetic; `scripts/math_sequential.py`.

5 Stratified guarantees for rare case types

Cases arrive in one stream; each carries a type $k \in \{1, \ldots, K\}$ (dispute family) observed before its outcome. For each type, $H_{0,k} : \mathbb{E}[X \mid \text{past}, \text{type} = k] \leq m_k$, tested by an e-process E_k that updates only on type-k cases.

Lemma 5.1 (Stratum e-processes on a shared stream)**.** *Under $H_{0,k}$, E_k is a non-negative supermartingale for the global filtration, so $\mathbb{P}(\sup_t E_{k,t} \geq a) \leq 1/a$ and $\mathbb{E}[E_{k,\tau}] \leq 1$ for every global stopping time τ.* *[adaptation]*

Proof. Let $\mathcal{F}'_{t-1}$ be the global past together with the type of case t. If case t has type k, the factor is $1 + \lambda Y$ with λ $\mathcal{F}_{t-1}$-measurable and $\mathbb{E}[1 + \lambda Y \mid \mathcal{F}'_{t-1}] \leq 1$ by the argument of Proposition 4.3 applied conditionally on the type; otherwise the factor is 1. Taking conditional expectations given $\mathcal{F}_{t-1}$ preserves the inequality, and Lemma 4.1 finishes. □

Proposition 5.2 (Time-uniform weighted e-Bonferroni)**.** *Let $w_k \geq 0$ with $\sum_k w_k = K$. Flag type k the first time $E_{k,t} \geq K/(\alpha w_k)$ (never if $w_k = 0$). Then $\mathbb{P}$(some true null is ever flagged) $\leq \alpha$, under any dependence between types.* *[standard]*

Proof. By Lemma 5.1 with $a = K/(\alpha w_k)$ and a union bound over the set I_0 of true nulls, the probability is at most $\sum_{k \in I_0} \alpha w_k / K \leq \alpha$. □

Proposition 5.3 (Closed testing with mean e-values)**.** *At a stopping time τ, reject $H_{0,k}$ iff $\frac{1}{|S|}\sum_{j \in S} E_{j,\tau} \geq 1/\alpha$ for every $S \ni k$. Then the family-wise error is at most α; every hypothesis*

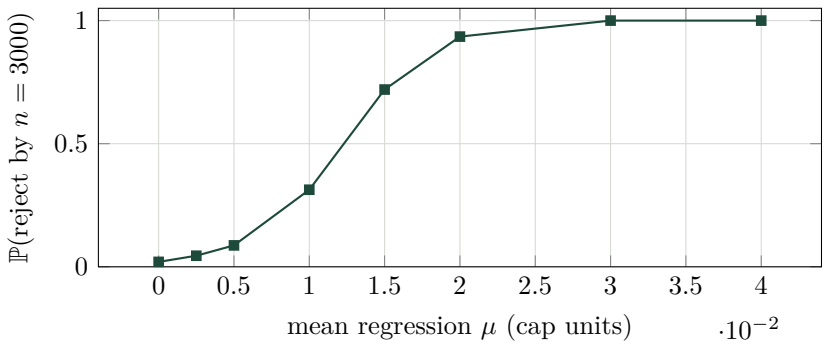


Figure 5: Power of the regression test (aGRAPA), ternary data with 10% of cases changed by ± 0.8, 1,000 runs per point. Synthetic.

rejected by unweighted e-Bonferroni at level α is rejected; and for each size s the minimum over $|S| = s$, $S \ni k$, is attained by k together with the $s - 1$ smallest other e-values.

[standard; Marcus et al., 1976; Vovk and Wang, 2021]

Proof. Let I_0 be the true nulls. The mean $\bar{E}_{I_0} = \frac{1}{|I_0|} \sum_{j \in I_0} E_{j,\tau}$ has $\mathbb{E}\bar{E}_{I_0} \le 1$ by Lemma 5.1, so $\mathbb{P}(\bar{E}_{I_0} \ge 1/\alpha) \le \alpha$ by Markov's inequality. A false rejection of some $k \in I_0$ requires the condition for $S = I_0 \ni k$, i.e. the event above. If $E_k \ge K/\alpha$ then for every $S \ni k$, $\frac{1}{|S|} \sum_S E_j \ge E_k/|S| \ge E_k/K \ge 1/\alpha$. For fixed size, the sum over S with $k \in S$ is E_k plus a sum of $s - 1$ others, minimised by the smallest ones. □

Proposition 5.4 (Weighted e-BH)**.** *At a stopping time, with $e'_k = w_k E_{k,\tau}$, reject the r largest e'_k, where r is the largest j with $e'_{(j)} \ge K/(\alpha j)$. Then* FDR $\le \alpha \sum_{k \in I_0} w_k/K \le \alpha$ *under any dependence.* *[standard; Wang and Ramdas, 2022]*

Proof. With R rejections, every rejected k satisfies $e'_k \ge K/(\alpha R)$, so $\mathbf{1}\{k \text{ rejected}\}/\max(R, 1) \le \alpha e'_k/K$. Summing over $k \in I_0$ and taking expectations, FDR $\le \frac{\alpha}{K} \sum_{k \in I_0} w_k \mathbb{E}[E_{k,\tau}] \le \frac{\alpha}{K} \sum_{k \in I_0} w_k$. □

Remark 5.5. Propositions 5.3 and 5.4 hold at one stopping time; "reject whenever e-BH has ever rejected" during monitoring is not covered. The gate uses Proposition 5.2 while monitoring.

Evidence (synthetic). Eight types with frequencies $(0.30, 0.20, 0.15, 0.12, 0.10, 0.08, 0.04, 0.01)$; the rare type regresses by $+0.35$ cap units while every other type improves by 0.01, so the pooled mean improves (-0.0064). Over 500 runs of 8,000 cases a pooled anytime test flagged a regression 0.006 of the time; stratified e-Bonferroni flagged the rare type 1.000 of the time. At the null boundary the old zero-tolerance "no slice mean may get worse" check false-blocked 0.73 to 0.996 of the time, the stratified procedures at most 0.03.

6 Planning and the cap tax

Normalise $m = t_k/c_k$, $d = \Delta_k/c_k$, $\sigma = \sigma_k/c_k$; let $Y = X - m$ with $\mathbb{E}Y = d$, $M_2 = \mathbb{E}Y^2 = \sigma^2 + d^2$, constant bet λ and $a = \lambda(1 + m) < 1$.

Lemma 6.1 (Growth bound)**.** *For $a \in [0, 1)$ and $x \ge -a$, $\log(1 + x) \ge x - \frac{x^2}{2(1-a)}$. Hence $g(\lambda) := \mathbb{E}\log(1 + \lambda Y) \ge \lambda d - \frac{\lambda^2 M_2}{2(1-a)}$.* *[adaptation]*

Proof. Let $f(x) = \log(1 + x) - x + \frac{x^2}{2(1-a)}$. Then $f(0) = 0$ and $f'(x) = x(\frac{1}{1-a} - \frac{1}{1+x})$. For $x \in [-a, 0]$, $1 + x \ge 1 - a$ so the bracket is ≥ 0 and $f' \le 0$; for $x \ge 0$ both factors are ≥ 0. So f is minimised at 0 and $f \ge 0$. Apply with $x = \lambda Y \ge -\lambda(1 + m) = -a$ and take expectations. □

Proposition 6.2 (Sufficient stratum sample size)**.** *Let $Z = \log(1+\lambda Y)$, $v = \lambda^2 M_2/(1-a)^2$, $b = \lambda d - \log(1-a)$, $l = \log(1/\beta)$ and $\varepsilon(n) = lb/3 + \sqrt{(lb/3)^2 + 2lnv}$. For i.i.d. observations, the constant-bet e-test at level α_k rejects by n observations with probability at least $1-\beta$ whenever*

$$n\, g(\lambda) - \varepsilon(n) \geq \log(1/\alpha_k). \tag{4}$$

If additionally the type count is $n_k \sim \mathrm{Binomial}(N, \pi_k)$, independent of the type-k outcomes, and $\mathbb{P}(n_k \geq n) \geq 1-\gamma$, the power with N total cases is at least $(1-\beta)(1-\gamma)$. [adaptation]

Proof. For $x \geq -a$, $|\log(1+x)| \leq |x|/(1-a)$: for $x \geq 0$, $\log(1+x) \leq x$; for $x \in [-a, 0)$, $-\log(1+x) \leq -x/(1+x) \leq |x|/(1-a)$. Hence $\mathrm{Var} Z \leq \mathbb{E}Z^2 \leq v$. By Jensen, $\mathbb{E}Z \leq \log(1+\lambda d) \leq \lambda d$, and $Z \geq \log(1-a)$, so $\mathbb{E}Z - Z \leq b$. Bernstein's inequality (Boucheron et al., 2013, Thm. 2.10 and Cor. 2.11) for $S_n = \sum_{i \leq n} Z_i$ gives $\mathbb{P}(n\mathbb{E}Z - S_n \geq \varepsilon) \leq \exp\big(-\frac{\varepsilon^2}{2(nv + b\varepsilon/3)}\big)$, and the stated $\varepsilon(n)$ solves $\varepsilon^2 = 2l(nv + b\varepsilon/3)$, so the right side is β. With probability at least $1-\beta$, $\log K_n = S_n \geq n\mathbb{E}Z - \varepsilon(n) \geq ng(\lambda) - \varepsilon(n) \geq \log(1/\alpha_k)$ by Lemma 6.1 and (4), and the test rejects. For the second claim, the test rejects if its wealth ever crosses, so its rejection probability given $n_k \geq n$ is at least its probability of crossing by the n-th type-k case, which is at least $1-\beta$ by independence of counts and outcomes. □

Proposition 6.3 (The cap tax: a lower bound for every valid test)**.** *Let $P_1 = \delta_{m+d}$ and $P_0 = (1-p)\delta_{m+d} + p\,\delta_{-1}$ with $p = d/(1+m+d)$. Every test valid at level α for all laws on $[-1, 1]$ with mean $\leq m$ and with power $1-\beta$ against P_1 after n i.i.d. observations satisfies*

$$n \;\geq\; \frac{\mathrm{kl}(1-\beta \| \alpha)}{\log\big(1 + \frac{d}{1+m}\big)}, \tag{5}$$

and a sequential test satisfies the same bound for $\mathbb{E}_{P_1}[\tau]$. [standard; Kaufmann et al., 2016]

Proof. P_0 has mean $(1-p)(m+d) - p = m$, so it is in the null and $P_0^n(A) \leq \alpha$ for the rejection event A, while $P_1^n(A) \geq 1-\beta$. $\mathrm{KL}(P_1 \| P_0) = \log\frac{1}{1-p} = \log\frac{1+m+d}{1+m}$, and $\mathrm{KL}(P_1^n \| P_0^n) = n\mathrm{KL}(P_1 \| P_0)$. Lemma 4.2 gives the bound. The sequential version replaces n by $\mathbb{E}_{P_1}[\tau]$ via the change-of-measure lemma of Kaufmann et al. (2016, Lemma 1). □

Because $d = \Delta/c$, the required sample size grows linearly with the declared cap: a loose cap is paid for in cases by every valid nonparametric test. On the synthetic illustration (cap \$100, regression \$5, s.d. \$20, 1% stratum, eight strata) the Bernstein plan needs 795 rare-stratum cases (simulated minimum about 341) and the optimised α allocation saves 28% to 35% of total traffic against uniform α/K.

7 Variance-adaptive betting and a floor from observed data

Paired cost differences are mostly zero with a few large values. The default bet of Algorithm 1 caps λ at half the largest valid bet, which binds when the variance is small relative to the mean, and its plug-in mean is noisy while most observations are zero. We compare four bets, all valid for the same null: *agrapa* (default), *agrapa_wide* (cap $0.95/(1+sm)$), *plugin* (the predictable plug-in bet of Waudby-Smith and Ramdas, 2024, $\lambda_t = \min\{0.95/(1+sm), \sqrt{2\log(1/a)/(\hat{\sigma}^2_{t-1} t \log(1+t))}\}$) and *mixture* (Algorithm 2).

Proposition 7.1 (Grid mixture)**.** *With $\lambda_j = \lambda_{\max}\rho^j$, $j < J$, $\lambda_{\max} \leq b/(1+sm)$, $b < 1$, the process $E_t = \frac{1}{J}\sum_j \prod_{i \leq t}(1+\lambda_j Y_i)$ is a non-negative supermartingale with $E_0 = 1$ under H_0, and on every path*

$$\log E_t \;\geq\; \max_j \log K_t(\lambda_j) - \log J. \tag{6}$$

If the Y_i are i.i.d. and $g(\lambda) = \mathbb{E}\log(1+\lambda Y)$ is maximised at $\lambda^\star \in [\lambda_{J-1}, \lambda_0]$, then $\liminf_t \frac{1}{t}\log E_t \geq \rho\, g(\lambda^\star)$ almost surely. [adaptation]

Algorithm 2 Grid mixture of constant bets

Require: m, s, α, $\lambda_{\max} = 0.95/(1+sm)$, ratio $\rho = 1/2$, grid size $J = 14$

$\lambda_j \leftarrow \lambda_{\max}\rho^j$ and $L_j \leftarrow 0$ for $j = 0, \dots, J-1$

for each case i **do**

  $Y \leftarrow s(X_i - m)$; $L_j \leftarrow L_j + \log(1 + \lambda_j Y)$ for all j

  $\log E \leftarrow \log\left(\frac{1}{J}\sum_j e^{L_j}\right)$ ▷ computed stably

  if $\log E \geq \log(1/\alpha)$ **then return** reject

  end if

end for

Proof. Each $K_t(\lambda_j)$ is a non-negative supermartingale with $K_0 = 1$ (Proposition 4.3 with a constant bet). A convex combination of non-negative supermartingales on the same filtration is a non-negative supermartingale, with initial value 1. A sum of non-negative terms is at least each term, so $E_t \geq K_t(\lambda_j)/J$ for every j, which is (6). For the rate: by the strong law, $\frac{1}{t}\log K_t(\lambda_j) \to g(\lambda_j)$ a.s. for each of the finitely many j (the summands are bounded below by $\log(1-b)$ and above by $\log 2$); with (6), $\liminf \frac{1}{t}\log E_t \geq \max_j g(\lambda_j)$. Some λ_j lies in $[\rho\lambda^\star, \lambda^\star]$. g is concave with $g(0) = 0$, so $g(\theta\lambda^\star) \geq \theta g(\lambda^\star)$ for $\theta \in [0,1]$, and $g(\lambda_j) \geq \rho g(\lambda^\star)$. □

Proposition 7.2 (The best constant-bet growth is a distance to the null)**.** *Let P be a law on $[-1,1]$ with mean above m, $Y = X - m$, $\bar\lambda = 1/(1+m)$, and $g(\lambda) = \mathbb{E}_P \log(1+\lambda Y)$ on $[0, \bar\lambda]$. Let $\lambda^\star$ maximise g with $g^\star = g(\lambda^\star) > 0$, and suppose either (i) $\lambda^\star < \bar\lambda$, or (ii) $\lambda^\star = \bar\lambda$ and $P([-1, -1+\varepsilon_0)) = 0$ for some $\varepsilon_0 > 0$. Define*

$$Q = \frac{1}{1+\lambda^\star Y}\,P + w\,\delta_{-1}, \qquad w = 1 - \mathbb{E}_P\Big[\frac{1}{1+\lambda^\star Y}\Big].$$

Then $w \geq 0$ ($w = 0$ in case (i)), Q is a probability on $[-1,1]$ with mean exactly m, and $\mathrm{KL}(P\|Q) = g^\star$. Consequently every test valid at level α for all laws on $[-1,1]$ with mean $\leq m$ and with power $1-\beta$ against P after n i.i.d. observations satisfies

$$n \;\geq\; \mathrm{kl}(1-\beta\|\alpha)/g^\star, \tag{7}$$

and a sequential test satisfies the same bound for $\mathbb{E}_P[\tau]$.

[adaptation; $g^\star = K_{\inf}$ is the duality of Honda and Takemura, 2010]

Proof. Write $c(\lambda) = \mathbb{E}_P[Y/(1+\lambda Y)]$. In case (i), on a neighbourhood $[0, \lambda^\star + \delta]$ with $\lambda^\star + \delta < \bar\lambda$, $1 + \lambda Y \geq 1 - (\lambda^\star + \delta)(1+m) > 0$, so $Y/(1+\lambda Y)$ is bounded and g is differentiable there with $g' = c$ (dominated convergence). g is concave, so an interior maximiser, or the maximiser $\lambda^\star = 0$ (excluded since $g^\star > 0 = g(0)$), has $c(\lambda^\star) = 0$. In case (ii), $1 + \bar\lambda Y = (1+X)/(1+m) \geq \varepsilon_0/(1+m)$ P-a.s., so g is differentiable on $[0, \bar\lambda]$ (one-sided at $\bar\lambda$) with $g' = c$, and a maximum at the right end forces $c(\bar\lambda) \geq 0$. In both cases $c(\lambda^\star) \geq 0$ and $\lambda^\star c(\lambda^\star)(1 - \lambda^\star(1+m)) = 0$.

Since $\frac{1}{1+\lambda Y} = 1 - \frac{\lambda Y}{1+\lambda Y}$, $\mathbb{E}_P\big[\frac{1}{1+\lambda^\star Y}\big] = 1 - \lambda^\star c(\lambda^\star)$, so $w = \lambda^\star c(\lambda^\star) \geq 0$, with $w = 0$ in case (i), and Q has total mass 1; its density with respect to P is positive, so Q is a probability. Its mean shift is

$$\mathbb{E}_Q[Y] = \mathbb{E}_P\Big[\frac{Y}{1+\lambda^\star Y}\Big] + w\cdot(-(1+m)) = c(\lambda^\star)(1 - \lambda^\star(1+m)) = 0,$$

using $c(\lambda^\star) = 0$ in case (i) and $\lambda^\star(1+m) = 1$ in case (ii). So $\mathbb{E}_Q X = m$ and Q lies in the null. P puts no mass at -1 in case (ii) and $w = 0$ in case (i), so on the support of P, $\frac{dP}{dQ} = 1 + \lambda^\star Y$ and $\mathrm{KL}(P\|Q) = \mathbb{E}_P \log(1+\lambda^\star Y) = g^\star$. For the rejection event A, $P^n(A) \geq 1-\beta$ and $Q^n(A) \leq \alpha$; Lemma 4.2 and $\mathrm{KL}(P^n\|Q^n) = n\,g^\star$ give (7). The sequential statement follows from Kaufmann et al. (2016, Lemma 1). □

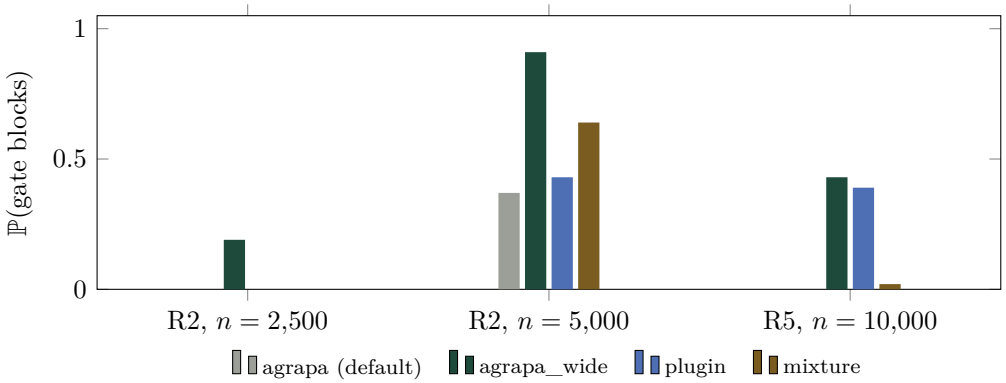


Figure 6: Power at equal validity on E1's planted regressions, cases resampled i.i.d. from the benchmark's paired costs (synthetic).

Remark 7.3. The converse inequality $\mathrm{KL}(P\|Q') \geq g^\star$ for every null Q' (so that $g^\star$ is exactly the smallest divergence to the null, $K_{\inf}$) follows from the Donsker–Varadhan variational formula with the test function $\log(1 + \lambda^\star Y)$; we only use the direction proved above. Applied to an empirical law $\hat{P}$, (7) answers: if cases were drawn i.i.d. from the paired costs we observed, how many would any valid test need? It is a statement about the observed distribution, not about a population.

Evidence on E1's planted regressions (synthetic). We replayed the E1 policies on all 10,000 benchmark cases and stored the paired merchant-cost differences (USD minor units, no case content; file `results/experiments/E1_paired_costs.json`). The gate is the E1 detector's block family (six families plus pooled, weighted e-Bonferroni on running maxima, cap \$200, $\alpha = 0.05$), with each betting strategy in turn. With the default bet it reproduces E1's verdicts exactly (R2 blocked on the pooled process; R3 on buyer's remorse, service failure and pooled; the rest held). Table 2 and Figure 6 summarise `scripts/math_betting.py analyse`.

Table 2: Betting strategies on E1's paired costs (synthetic). Top: largest running-max $\log E$ over the seven gate hypotheses on the full benchmark (rejection needs $\log(7/0.05) = 4.94$). Bottom: $\mathbb{P}$(gate blocks) when n cases are resampled i.i.d. from the benchmark's paired costs, 100 resamples per cell. All four bets are valid for the same null; at the null boundary (centred R5 and R3 costs, 400 runs of 2,000) they rejected 0 to 0.0025 of the time.

		agrapa (default)	agrapa_wide	plugin	mixture
$\log E$, full benchmark	R1	2.13	3.54	2.84	1.54
	R2	9.91	16.62	7.93	14.23
	R3	34.42	58.78	15.84	56.71
	R4	1.96	3.56	3.66	1.54
	R5	2.91	**5.11**	4.62	2.80
$\mathbb{P}$(block), resampled	R2, $n = 2{,}500$	0.00	0.19	0.00	0.00
	R2, $n = 5{,}000$	0.37	0.91	0.43	0.64
	R1, $n = 10{,}000$	0.00	0.17	0.00	0.01
	R4, $n = 10{,}000$	0.00	0.04	0.01	0.00
	R5, $n = 10{,}000$	0.00	0.43	0.39	0.02

What the numbers say:

- **The default bet's cap is the bottleneck.** In almost every regressed family the best constant bet on the observed costs is the largest valid one, $\lambda^\star = 1/(1 + m)$ (case (ii) of Proposition 7.2): the differences are zero on most cases and large and positive on a few,

so the Kelly bet is everything. The default stops at half of it. Raising the cap to 0.95 multiplies the final log-wealth by 1.66 to 1.82 and lifts R2's resampled power at 5,000 cases from 0.37 to 0.91.

- **R5, the regression E1 missed, is visible to a stratified test, not to a pooled one.** Its pooled mean is 1.34 USD minor per case, about one cent ($t = 0.27$), and Proposition 7.2 says any valid pooled test needs at least 509,153 cases for power 0.8 at $\alpha = 0.05$. But its cost is concentrated: in buyer's remorse disputes the mean (after clipping at the cap) is 64.35 minor per case, and the floor there is 689 family cases, about 3,486 benchmark cases in total (6,316 at the gate's per-hypothesis level 0.05/7). On the actual benchmark the wide bet blocked R5 on buyer's remorse ($\log E = 5.11$); on resampled benchmarks of 10,000 cases it did so 43% of the time, the plug-in bet 39%, the default never.
- **The mixture pays for its grid.** Its log-wealth is within $\log 14 = 2.64$ of the best grid bet (Proposition 7.1), which costs it detections when the best bet is the largest one. It wins only when $\lambda^\star$ is far from the cap, which these regressions are not.
- **Nothing helps on 500-case folds.** No strategy blocked any of the 20 folds of any policy; the floors above are thousands of cases.

These are statements about the benchmark's own paired costs (resampling reuses them), not about a population of agent changes. A variance-reduced gate built since was tested on the same regressions in a pre-registered experiment (Section 11.3); nothing in this section depends on it.

8 Asymmetric costs

Let θ be the true mean per-case loss difference and δ the tolerance. A wrong approval (promote while $\theta > \delta$) costs c_A; a wrong block costs c_B.

Proposition 8.1 (Bayes rule)**.** *Under a posterior Π, the rule that minimises posterior expected 0–1 loss promotes iff $\Pi(\theta > \delta) \le c_B/(c_A + c_B)$.* *[standard; Berger, 1985]*

Proof. Posterior risk of promoting is $c_A\Pi(\theta > \delta)$ and of blocking $c_B\Pi(\theta \le \delta) = c_B(1 - \Pi(\theta > \delta))$; promote iff the first is at most the second, which rearranges to the threshold. Bayes risk is the integral of posterior risk over the marginal of the data (Fubini), so the pointwise minimiser minimises it. □

Proposition 8.2 (Composition with the certificate)**.** *Let E be an e-value (or a running maximum used with Ville) for $H_0 : \theta \ge \delta$ and $A = \{E \ge 1/\alpha\}$. The composed rule promotes iff $x \in A$ and the Bayes rule promotes. Then (1) $\mathbb{P}_\theta(\text{promote}) \le \alpha$ for every $\theta \ge \delta$, whatever the prior and whether or not the posterior model is right; (2) among all rules $\varphi \le \mathbf{1}_A$ it minimises Bayes risk under the model.* *[adaptation; restricted Bayes in the sense of Hodges and Lehmann, 1952]*

Proof. (1) $\{\text{promote}\} \subseteq A$ and $\mathbb{P}_\theta(A) \le \alpha$ by Markov's inequality or Lemma 4.1. (2) Bayes risk equals $\int[\varphi(x)r_P(x) + (1 - \varphi(x))r_B(x)]\, dm(x)$, with r_P, r_B the posterior risks of promoting and blocking. Under the constraint $\varphi \le \mathbf{1}_A$, the integrand is minimised pointwise by $\varphi(x) = \mathbf{1}\{x \in A, r_P(x) \le r_B(x)\}$, which is the composed rule. □

Synthetic evidence: at 1:1 costs the Bayes rule alone promotes 0.614 of boundary-harmful changes; composed with the certificate, 0.0017.

9 What the whole promote decision guarantees

The gate promotes iff every conjunct passes: hard checks H on every case, a soft-score threshold S, no block from the block family B (Proposition 5.2), and the certificate T (Proposition 4.3

with $s = -1$). A *harm* is a property of the candidate we never want to ship. A conjunct c *covers* harm j at level $\alpha_{c,j}$ if $\sup\{\mathbb{P}(c \text{ passes}) : \text{candidate has harm } j\} \le \alpha_{c,j}$; a deterministic check that reads the harm on every case has level 0, and a sampled audit has the level of Theorem 10.3.

Theorem 9.1 (Composition)**.** *(a) (Promote side.) For a candidate with harms J,*

$$\mathbb{P}(\textit{promote}) \le \min_{j \in J} \min_{c} \alpha_{c,j},$$

and the bound is 1 if no conjunct covers any harm in J.

(b) (Block side.) If the candidate has no harm and its hard checks raise a false alarm on case i with probability ϕ_i, then $\mathbb{P}(\textit{block}) \le \alpha_{\textit{block}} + \mathbb{P}(\exists i : \textit{false alarm on } i)$, which equals $\alpha_{\textit{block}} + 1 - \prod_i (1 - \phi_i)$ under independence and is at most $\alpha_{\textit{block}} + \sum_i \phi_i$ in general.

(c) (Resubmission.) If submission k of a candidate line is certified at level α_k with $\sum_k \alpha_k \le \alpha$, the probability that any harmful submission is ever promoted is at most α.

[standard, applied; intersection-union principle of Berger, 1982; spending in the spirit of Foster and Stine, 2008]

Proof. (a) $\{\text{promote}\} \subseteq \{c \text{ passes}\}$ for every conjunct c, so $\mathbb{P}(\text{promote}) \le \mathbb{P}(c \text{ passes}) \le \alpha_{c,j}$ for each covering pair; take the minimum. (b) $\{\text{block}\} \subseteq \{B \text{ fires}\} \cup \{\text{some hard check fires}\}$; the first has probability at most α_{block} for a harmless candidate (Proposition 5.2); the second is $1 - \prod_i (1 - \phi_i)$ under independence and at most $\sum_i \phi_i$ by a union bound. (c) The event is the union over k of "submission k is harmful and promoted", each of probability at most α_k by (a). □

Corollary 9.2 (Per-type certificates)**.** *Run one certificate e-process per case type for $H_{0,k} : \mu_k \ge t_k$ on the interleaved stream and promote only if every type is certified at level α. If any one type is harmful, $\mathbb{P}(\textit{promote}) \le \alpha$, with no division of α by K.* *[adaptation]*

Proof. If type k is harmful, "type k certified" is a conjunct covering that harm at level α by Lemma 5.1; apply Theorem 9.1(a). □

What is not guaranteed. (1) Harms no conjunct covers: E2's cheaper fraud (Section 11.2) makes the null "cost is worse" false, so B and T say nothing; E4's unfair wins are scored by the soft OUT-01. (2) Rare-type regressions under the shipped *pooled* certificate (Figure 7). (3) Soft scores and the Bayes rule can only remove promotions, but carry no guarantee of their own. (4) Level 0 for a hard check assumes the check is correct; the verifier and the simulator share a world model. (5) Shift between held-out and live traffic, and delayed outcomes in canary.

Evidence (synthetic). With one common type exactly at its tolerance, per-type certificates promoted at most 0.02 of the time ($\alpha = 0.05$). A hard check with false-alarm rate $\phi = 0.0019$ per case (the honest policy's measured INV-01 rate) blocked an honest candidate 0.319 of the time at $n = 200$, 0.596 at 500, 0.979 at 2,000 and 1.000 at 10,000 (bounds 0.366, 0.664, 1.000, 1.000). Twenty harmful versions at a fixed $\alpha = 0.05$: one was promoted in 0.402 of runs; with spending $\alpha_k = 6\alpha/(\pi^2 k^2)$, in 0.017.

10 Hacking-gap bounds

Definition 10.1 (Hacking gap)**.** On case i the candidate has a proxy score P_i (a self-report or a naive win label) and an anchored score A_i (recomputed from the ledger), in cap units, with $0 \le D_i = P_i - A_i \le \varepsilon$. The hacking gap is $\Gamma = \frac{1}{n} \sum_{i \le n} D_i$. Fix a tolerance $\eta \in [0, \varepsilon]$; case i is *hacked* if $D_i > \eta$, $h_i = \mathbf{1}\{D_i > \eta\}$ and $M_t = \sum_{i \le t} h_i$. Some cases are *flagged*; the batch is *accepted* iff no case is flagged.

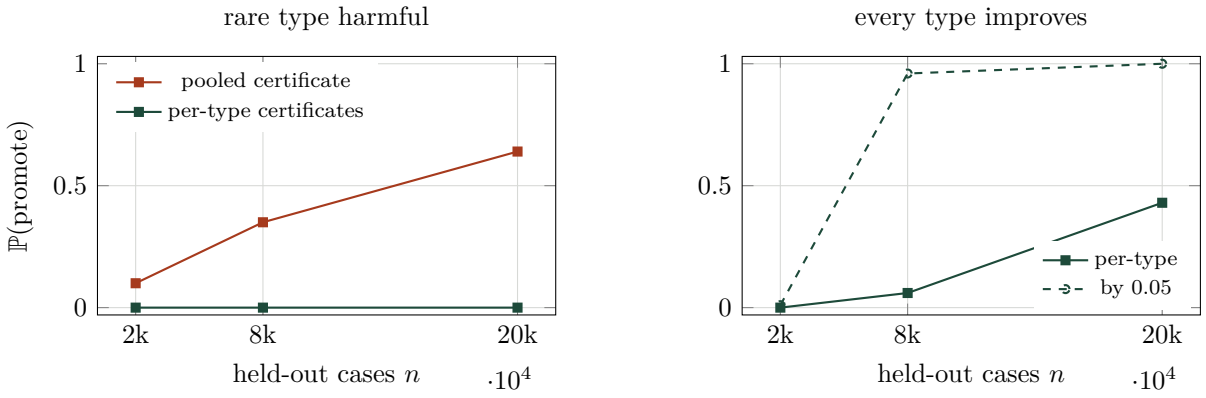


Figure 7: Left: a 1% type regresses by 0.35 cap units and the rest improve by 0.01. The pooled certificate promotes more often as n grows, because the pooled mean really is below tolerance; per-type certificates never did. Right: the power cost of per-type certificates (the pooled certificate promotes 1.00 in both settings). 100 runs per point; synthetic; `scripts/math_composition.py`.

Assumption 10.2 (Catch floor)**.** Cases are handled in order. Before case i the agent holds information $\mathcal{G}_i \supseteq \mathcal{F}_{i-1}$ (the history, any signal about case i, its own randomness) and chooses D_i as a $\mathcal{G}_i$-measurable variable. Let $\mathcal{H}_i = \sigma(\mathcal{G}_i)$ and Fl_i the event that case i is flagged, with $\mathrm{Fl}_i \in \mathcal{F}_i$. There is $\rho \in [0,1]$ with

$$\mathbb{P}(\mathrm{Fl}_i \mid \mathcal{H}_i) \;\geq\; \rho \quad \text{on } \{h_i = 1\}.$$

Assumption 10.2 allows randomised and adaptive policies, audits that depend on the past, an agent that sees past audit results or partial signals, and correlated detections; flags on honest cases are allowed.

Theorem 10.3 (Gap of an accepted change, adaptive agents)**.** *Under Assumption 10.2, with* $m_0 = \lceil n(\gamma - \eta)/\varepsilon \rceil$,

$$\mathbb{P}(\textit{accepted and } \Gamma \geq \gamma) \leq (1-\rho)^{m_0}, \tag{8}$$

$$\mathbb{E}[\Gamma\, \mathbf{1}\{\textit{accepted}\}] \leq \eta + \frac{\varepsilon}{n\, e\, \ln \frac{1}{1-\rho}}. \tag{9}$$

[adaptation of the survival argument to a supermartingale; framing candidate novel]

Proof. Write $r = 1 - \rho$; if $\rho = 0$ both bounds are trivial ($\Gamma \leq \varepsilon$), and if $\rho = 1$ no hacked case is accepted, so assume $0 < \rho < 1$. Let $A_t = \bigcap_{i \leq t} \mathrm{Fl}_i^c$ and $Z_t = \mathbf{1}_{A_t} r^{-M_t}$, $Z_0 = 1$. Then $Z_i = Z_{i-1}\, r^{-h_i} \mathbf{1}_{\mathrm{Fl}_i^c}$, and Z_{i-1} and h_i are $\mathcal{H}_i$-measurable, so

$$\mathbb{E}[Z_i \mid \mathcal{H}_i] = Z_{i-1} r^{-h_i}\, \mathbb{P}(\mathrm{Fl}_i^c \mid \mathcal{H}_i) \leq Z_{i-1} :$$

if $h_i = 1$, $\mathbb{P}(\mathrm{Fl}_i^c \mid \mathcal{H}_i) \leq 1 - \rho = r$ by Assumption 10.2; if $h_i = 0$ the factor is at most 1. By the tower property ($\mathcal{F}_{i-1} \subseteq \mathcal{H}_i$), $\mathbb{E}[Z_i] \leq \mathbb{E}[Z_{i-1}]$ and $\mathbb{E}[Z_n] \leq 1$. On A_n, $Z_n = r^{-M_n}$, so

$$\mathbb{P}(A_n \cap \{M_n \geq m_0\}) = \mathbb{E}[Z_n\, r^{M_n} \mathbf{1}\{M_n \geq m_0\}] \leq r^{m_0} \mathbb{E}[Z_n] \leq r^{m_0}.$$

Since $D_i \leq \eta$ for unhacked and $D_i \leq \varepsilon$ for hacked cases, $\Gamma \leq \frac{(n-M_n)\eta + M_n \varepsilon}{n} \leq \eta + \frac{\varepsilon M_n}{n}$, so $\Gamma \geq \gamma$ forces $M_n \geq n(\gamma - \eta)/\varepsilon$, i.e. $M_n \geq m_0$; this proves (8). For (9), $\mathbb{E}[M_n \mathbf{1}_{A_n}] = \mathbb{E}[Z_n M_n r^{M_n}] \leq \sup_{x \geq 0} x r^x \cdot \mathbb{E}[Z_n]$. The function $x \mapsto x r^x$ has derivative $r^x(1 + x \ln r)$, vanishing at $x = 1/\ln(1/r)$ where its value is $1/(e \ln(1/r))$. Hence $\mathbb{E}[\Gamma \mathbf{1}_{A_n}] \leq \eta + \frac{\varepsilon}{n} \mathbb{E}[M_n \mathbf{1}_{A_n}] \leq \eta + \frac{\varepsilon}{ne \ln(1/r)}$. □

Corollary 10.4 (Independent audits)**.** *If each case is audited independently with probability q, independently of the agent's information, and an audited hacked case is flagged with probability at least $1-\beta$ given the agent's information, then Theorem 10.3 holds with $\rho = q(1-\beta)$. For $\beta = 0$ this is the earlier spot-check bound: $\mathbb{E}[\Gamma \mathbf{1}\{acc\}] \le \eta + \varepsilon/(ne \ln \frac{1}{1-q}) \approx \eta + \varepsilon/(enq)$.*

Proof. $\mathbb{P}(\mathrm{Fl}_i \mid \mathcal{H}_i) \ge \mathbb{P}(\mathrm{audited}_i \mid \mathcal{H}_i)(1-\beta) = q(1-\beta)$ on $\{h_i = 1\}$. □

Corollary 10.5 (Leaky audits)**.** *Let case i be audited ($a_i = 1$) with probability q, independently of $\mathcal{F}_{i-1}$, let the agent observe a signal S_i with $\mathbb{P}(S_i = s \mid a_i = 1, \mathcal{F}_{i-1}) \ge \ell\, \mathbb{P}(S_i = s \mid a_i = 0, \mathcal{F}_{i-1})$ for every s and some $\ell \in [0,1]$, let the agent's choice depend only on $(\mathcal{F}_{i-1}, S_i)$ and independent randomness, and let an audited hacked case be flagged with probability at least $1-\beta$ given $\mathcal{H}_i$ and $a_i = 1$. Then Assumption 10.2 holds with*

$$\rho = (1-\beta)\,\frac{q\ell}{q\ell + 1 - q}. \tag{10}$$

The bound is attained: an agent that sees "audited" with probability $1-\ell$ on audited cases, sees "not audited" otherwise, and hacks only on "not audited", has conditional catch probability exactly ρ on every hacked case when $\beta = 0$. *[adaptation]*

Proof. By Bayes, $\mathbb{P}(a_i = 1 \mid S_i = s, \mathcal{F}_{i-1}) = \frac{qL(s)}{qL(s)+1-q}$ with $L(s) = \frac{\mathbb{P}(s \mid a_i=1, \mathcal{F}_{i-1})}{\mathbb{P}(s \mid a_i=0, \mathcal{F}_{i-1})} \ge \ell$, and $x \mapsto \frac{qx}{qx+1-q}$ is non-decreasing. Independent randomness does not change this posterior, so $\mathbb{P}(a_i = 1 \mid \mathcal{H}_i) \ge \frac{q\ell}{q\ell+1-q}$ and $\mathbb{P}(\mathrm{Fl}_i \mid \mathcal{H}_i) \ge (1-\beta)\mathbb{P}(a_i = 1 \mid \mathcal{H}_i)$ on $\{h_i = 1\}$. For the attaining agent, the signal "not audited" has $L = \ell/1 = \ell$, so the posterior equals $q\ell/(q\ell + 1 - q)$ exactly. □

Theorem 10.6 (Minimal audit rate with an imperfect verifier)**.** *Cases are audited independently at rate q; an audited hacked case is flagged with probability $1-\beta$ and an audited honest case with probability ϕ, independently, with $q\phi < 1$. An accepted candidate pays the optimiser $V + \sum_{i \in S} g_i$ (honest value $V \ge 0$ plus a gain $0 \le g_i \le g$ per hacked case in the set S), a rejected one pays $-C$ ($C \ge 0$), and $K = V + C > 0$. Let $\mathrm{EU}(S)$ be the expected payoff of hacking S. Then $\mathrm{EU}(S) \le \mathrm{EU}(\emptyset)$ for every S if*

$$D := (1-\beta)(K+g) - \phi K > 0 \quad \text{and} \quad q \;\ge\; q^\star := \frac{g}{D}, \tag{11}$$

and conversely, if some case offers gain exactly $g > 0$ and $\mathrm{EU}(S) \le \mathrm{EU}(\emptyset)$ for every S, then (11) holds.

[candidate novel (framing); the deterrence condition goes back to Becker, 1968; Avenhaus et al., 2002]

Proof. With $|S| = m$, $\mathbb{P}(\mathrm{acc} \mid S) = (1-q\phi)^{n-m}(1-q(1-\beta))^m = (1-q\phi)^n s^m$ with $s = \frac{1-q(1-\beta)}{1-q\phi}$. Hence $\mathrm{EU}(S) + C = (1-q\phi)^n s^m (K + \sum_S g_i) \le (1-q\phi)^n f(m)$ with $f(m) = s^m(K + mg)$, and $\mathrm{EU}(\emptyset) + C = (1-q\phi)^n f(0)$. Now

$$\frac{f(m+1)}{f(m)} = s\Big(1 + \frac{g}{K+mg}\Big) \le s\Big(1 + \frac{g}{K}\Big),$$

so f is non-increasing whenever $s(K+g) \le K$. This inequality is $(1-q(1-\beta))(K+g) \le K(1-q\phi)$, i.e. $g \le q[(1-\beta)(K+g) - \phi K] = qD$, which under $D > 0$ is $q \ge g/D$. This proves sufficiency. Conversely, take $S = \{i\}$ with $g_i = g$: $\mathrm{EU}(\{i\}) \le \mathrm{EU}(\emptyset)$ is $(1-q\phi)^{n-1}(1-q(1-\beta))(K+g) \le (1-q\phi)^n K$; dividing by $(1-q\phi)^{n-1} > 0$ gives $g \le qD$, which with $g > 0$ forces $D > 0$ and $q \ge g/D$. □

Corollary 10.7 (When no audit rate deters)**.** *Under the conditions of Theorem 10.6, some $q \in [0,1]$ deters every hack iff $\beta g \le (1-\beta-\phi)K$, and $q^\star$ is increasing in β and in ϕ. If instead $K = 0$ (no honest value and nothing lost when caught, as with free resubmission), hacking one case is unprofitable only if $q(1-\beta) = 1$: spot checks cannot deter an agent that may resubmit for free.*

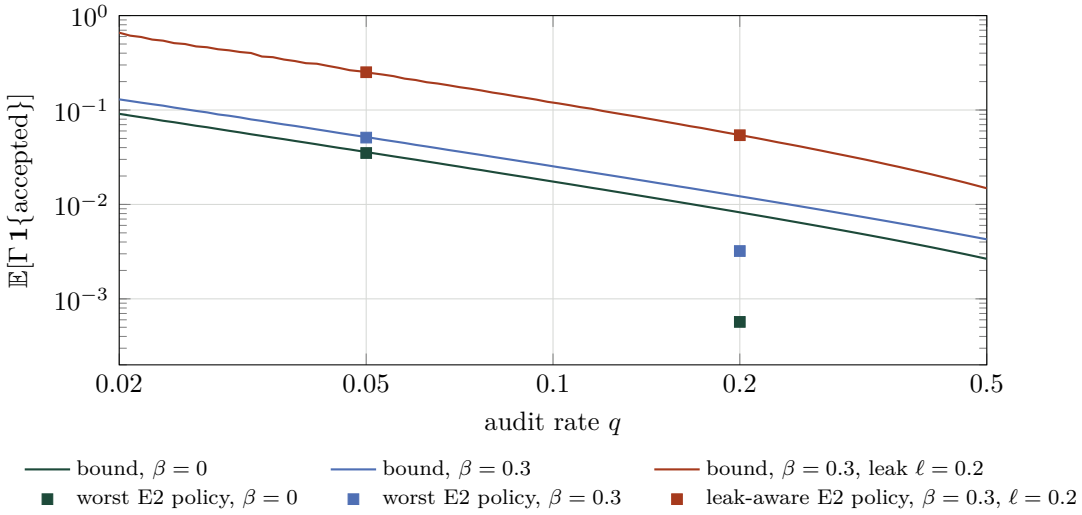


Figure 8: The expected accepted gap against the audit rate. Lines: Theorem 10.3 with ρ from Corollary 10.5, $n = 200$, $\varepsilon = 1$, $\eta = 0$, $\phi = 0$. Squares: the largest exact $\mathbb{E}[\Gamma\mathbf{1}\{\text{accepted}\}]$ over E2's 120 (policy, batch) pairs at $q \in \{0.05, 0.2\}$. A blind agent sits under the bound; at $q = 0.2$ far under it, because E2's policies hack 24 to 149 cases per batch and are almost always caught. The leak-aware agent attains the bound at $q = 0.05$ (0.25129 against 0.25134). Synthetic; `scripts/math_hacking_general.py`, `docs/math/05_hacking_gap.md`.

Proof. A deterring $q \le 1$ exists iff $g \le D$ (the condition $g \le qD$ is easiest at $q = 1$), and $g \le (1-\beta)(K+g) - \phi K$ rearranges to $\beta g \le (1-\beta-\phi)K$. D decreases in β and in ϕ, so $q^\star = g/D$ increases. If $K = 0$, $\mathrm{EU}(\{i\}) + C = (1-q\phi)^{n-1}(1-q(1-\beta))g$, which exceeds $\mathrm{EU}(\emptyset) + C = 0$ unless $1 - q(1-\beta) = 0$. □

Remark 10.8. $q^\star$ does not depend on n, but the honest acceptance probability $(1-q\phi)^n$ does: at $n = 10{,}000$, $q = 1$ and $\phi = 0.0019$ it is about 5.5×10^{-9}. False alarms are a tax on honest submissions.

Evidence on E2's policies (synthetic, exact). Experiment E2 (Section 11.2) recorded the number of hacked cases (false win claims) for 12 policies, every policy its proxy climb visited, on 10 batches of 200 benchmark cases: 120 batches, 110 with hacks (between 24 and 149 per batch). We replayed them under audits with $q \in \{0.05, 0.2\}$, $\beta \in \{0, 0.3\}$, $\ell \in \{1, 0.5, 0.2\}$ and $\phi \in \{0, 0.0019\}$, computing every probability exactly by binomial sums (and checking one setting against a seeded simulation: $\mathbb{P}(\text{acc})$ simulated 0.1454, exact 0.1439, s.e. 0.0025). Every tail (440 checks per setting) and every expectation lay under Theorem 10.3. With leakage the bound is attained to three decimals; the bound that ignores the leak is broken (Figures 8 and 9). On the E2 proxy a false win claim is worth $g = 1/(2n) = 0.0025$ against an honest value $V = 0.449$, so $q^\star = 0.00554$ for a perfect verifier, 0.00738 at $\beta = 0.25$ and 0.01107 at $\beta = 0.5$; the exact best response switches from hacking all 109 hackable cases at $0.5q^\star$ to 2 at $0.99q^\star$ and to none at $1.01q^\star$.

11 Experiments

11.1 E1: planted regressions (synthetic; pre-registered; pass, narrowly)

Seven regressions were planted in the evidence policy (R1 missed deadlines on 5% of a rare family; R2 a 2% queue stall; R3 wrong documents on Visa 13.1; R4 over-refunds below $25; R5 conceding every dispute below $25; R6 never requesting the cardholder statement; R7 claiming

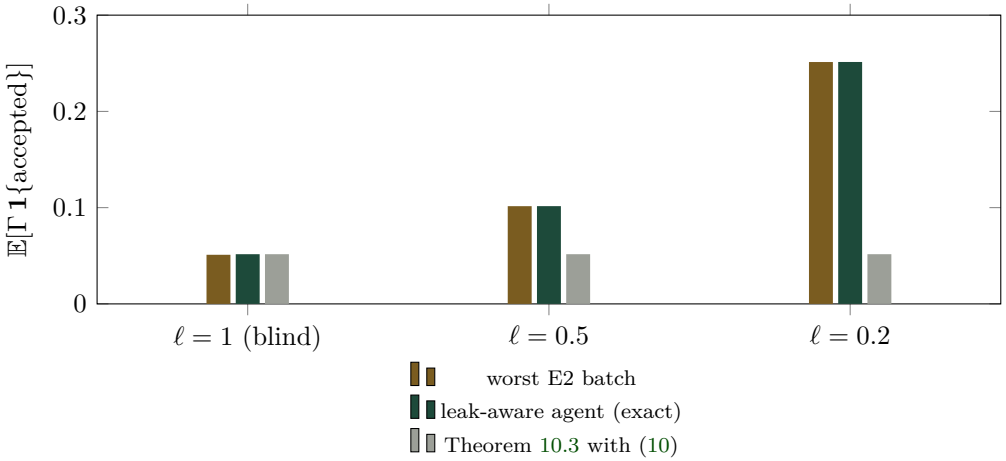


Figure 9: Leaky audits on E2's 120 batches, $n = 200$, $q = 0.05$, $\beta = 0.3$, $\phi = 0$. Synthetic; `scripts/math_hacking_general.py`.

every representment won) together with four controls. Each candidate was compared with the baseline on all 10,000 cases and on 20 folds of 500. The detector flags a new hard failure, a newly failing batch check, or a block by the statistical gate (cap \$200, $\alpha = 0.05$, families as types).

Table 3: E1 pre-registered criteria. Synthetic.

criterion	result	bar	holds
P1 all regressions flagged on the full benchmark	7 of 7	7 of 7	yes
P2 exercised folds flagged, pooled	114/135 = 0.844 (Wilson 0.774–0.896)	≥ 0.80	yes (point)
P3 control folds flagged	3/60 = 0.050 (0.017–0.137)	≤ 0.05	at the bar

Hard checks caught every regression that leaves a trace a check reads, on every fold where it happened. Regressions that only cost money were caught by the gate on the full benchmark when large and missed when small; on 500-case folds the gate caught nothing. The three control false alarms were INV-01 charging a refund race that stood at close to the agent; the cause turned out to be an engine bug, not the agent (Section 14). Section 7 re-analyses the paired costs with variance-adaptive bets.

A units correction. The E1 write-up gives mean cost differences of "\$188", "\$60" and "\$1.30" per case for R3, R2 and R5. The detector computes them in USD *minor* units: they are \$1.88, \$0.60 and \$0.013 per case. The planning statements of E1 were in minor units and are unaffected.

11.2 E2: an optimiser on a naive proxy (synthetic; pre-registered; fails H2)

A seeded coordinate ascent over 384 policies (8 knobs, 6 of them exploits) optimised either a proxy (mean of engine win rate and self-reported win rate) or the scorecard reward, on 150 public training episodes; the optima then played all 10,000 benchmark cases (Table 4).

Table 4: E2 on the full benchmark. Synthetic.

policy	reward	cost/case	hard-failure rate	engine win rate	correct
honest (= scorecard optimum)	0.966	\$82.82	0.19%	0.449	0.777
proxy optimum (forge, claim wins)	0.437	\$48.95	55.4%	0.795	0.557

H1 held (the proxy optimum uses two exploits, the scorecard optimum none), H3 held (the proxy optimum is flagged on 20 of 20 folds by EVD-01, SR-01 and PROC-01), and H4 held

(1,440 of 1,440 tail and 360 of 360 expectation checks of the gap bounds). **H2 failed**: the proxy optimum was $33.87 per case *cheaper* on the ledger (paired difference −33.87, 95% interval −35.64 to −32.10). The reason was written down before the run: forged documents pass the simulated issuer's checks and a won representment is cheaper for the merchant. The money saved is money the cardholder was owed, kept by forgery. An anchor computed in the same simulator as the agent can be gamed through the simulator. The cost test said promote; provenance and self-report checks blocked. This is exclusion (1) of Section 9, measured.

11.3 E3: a variance-reduced gate (synthetic; pre-registered; fails H1)

E1's gate held on R5 ("concede every dispute under $25") even on all 10,000 cases. E3 asked whether a variance-reduced gate detects it while keeping control false blocks at most 5%. The new test adds a *stratified* betting e-process, one wealth process whose bet depends on the case family, $K_t = \prod_{i \le t}(1 + \lambda_{k_i,i}(X_i - m_{k_i}))$, which is a non-negative supermartingale under the conditional null "no family regressed" by the argument of Lemma 5.1. It spends 0.8α (block at $K \ge 25$) and E1's block family spends 0.2α. Both settings were fixed from synthetic Monte Carlo before any benchmark case was played. Pre-registered: **H1** R5 is blocked on the full benchmark; **H2** no control is blocked; **H3** at most 5% of the 60 control folds are blocked. E3 passes iff all three hold.

A prediction against ourselves. Every e-value for the conditional null satisfies $\mathbb{E}_1[\log E] \le B = \sum_i \mu_i^+ / c$ (`docs/math/09`, Proposition 9.7, an instance of Proposition 6.3 applied case by case), and power 0.5 at level 0.04 needs $B \ge \mathrm{kl}(0.5 \| 0.04) = 0.94$. From E1's published means R5's pooled budget is 0.67, so the pre-registration predicted that H1 would fail: detecting R5 needs a tighter cap or more cases, not a better test.

Result: FAIL, as predicted, but closer. The replay reproduced E1's verdicts exactly. R5's stratified e-value reached 14.9 against the bar of 25; its family budget on the benchmark was 8.1 (pooled 0.67), because its cost sits in buyer's remorse while other families improve. H2 held (every control's stratified e-value was 1.0) and H3 held (0 of 60 folds, Wilson 95% upper bound 0.060). R2 and R3 stayed blocked (e-values 939 and 3.0×10^{14}); R1 (2.32) and R4 (3.47) held; no fold of any policy was blocked by either test.

Common random numbers, measured. Both policies see the same case and the same issuer draw. For R5 the variance of the paired difference was 2.4×10^5 (USD minor squared) with the same draws, 4.9×10^7 with independent issuer draws on the same cases, and 2.3×10^8 for two independent samples: same-draw pairing cuts the variance about 200-fold against same-case pairing alone and about 950-fold against unpaired samples. 92.8% of R5's differences are exactly zero with the same draws, 76.3% with independent draws.

Post hoc, not pre-registered: the bet cap. After the result, the same saved costs were re-analysed with the bet cap of Section 7 raised from 0.5 to 0.95 (still valid). The wide stratified test blocks R5 on the benchmark itself ($e = 119$ against 25). On 100 resamples of 10,000 cases (Figure 10) stratification and the wider cap are each worth about as much, and together detect R5 in 0.86 of resamples; no control was ever blocked. These resamples reuse the benchmark's own costs and were chosen after seeing the result: they motivate a next pre-registered experiment with the wide cap as primary. They are not a pass for E3.

11.4 Real data: CFPB complaints (real)

Three tasks on public-domain CFPB complaint narratives with frozen splits, graded by code against real fields (Table 5). Claude Haiku and Claude Sonnet answered held-out items blind,

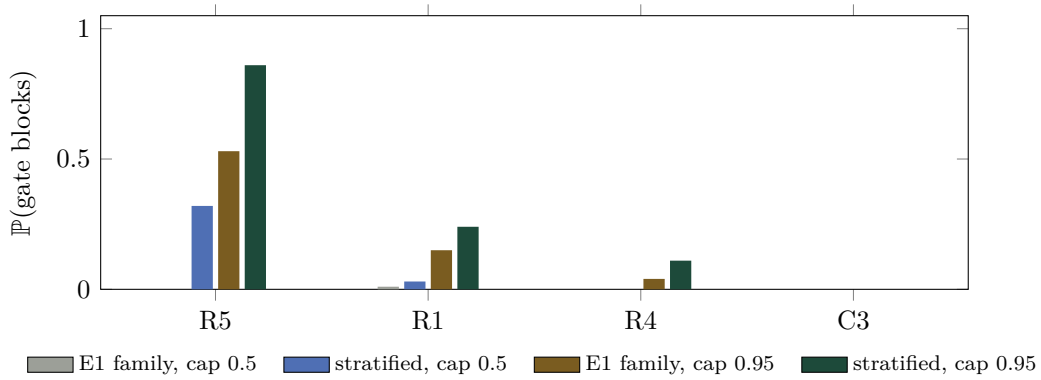


Figure 10: E3, post hoc and not pre-registered: share of 100 resampled benchmarks of 10,000 cases (seed 0) on which the gate blocks, for three cost-only regressions and the control C3. R5: 0, 0.32, 0.53, 0.86; R1: 0.01, 0.03, 0.15, 0.24; R4: 0, 0, 0.04, 0.11; C3: 0 throughout. Synthetic; `scripts/exp_e3_variance.py resample`, `results/experiments/E3_variance_reduction/posthoc_resampled.json`.

run as Claude Code subagents with the same prompt text as the API runner.

Table 5: Baselines on the full test splits (95% bootstrap intervals) and models on the same held-out items. Real data.

task	test n	majority	keywords	TF-IDF+LR	items	Haiku	Sonnet
outcome-2015 (AUROC)	2,975	0.500	0.579	0.722	100	0.550	0.606
regime-2015 (accuracy)	764	0.572	0.783	0.882	100	0.900	0.930
routing-2025 (issue acc.)	1,598	0.553	0.214	0.631	50	0.560	0.540

On the models' items TF-IDF scored 0.701, 0.870 and 0.580. Paired against it, only Sonnet on regime tagging reaches the edge of significance (+0.060, 95% interval 0.000 to 0.120); on outcome prediction both trail (Haiku -0.150, $[-0.318, 0.008]$; Sonnet -0.095, $[-0.213, 0.025]$). The models know the rules and misjudge which complaints end in relief.

11.5 Real models playing simulated disputes

Claude Haiku, Sonnet and Opus each handled the same 12 simulated disputes end to end through the episode command-line interface, briefed to play honestly and minimise the merchant's net loss (Figure 11). Sonnet was cheapest, $-$\$32.77 per case against the evidence policy (95% interval -64.01 to -4.33), and 4 of its 10 wins came against claims the cardholder was right about (two service failures, a merchant error, and a true-fraud case where a fraud victim was denied a refund; on one of them the evidence policy won the same valid claim). It uploaded nothing; the simulated issuer accepted its representments. OUT-01 marks the wins as unfair, but OUT-01 is soft, so they did not block. By Theorem 9.1(a) the gate carries no guarantee against this harm until a conjunct covers it.

Verifier false positives found by real models. Real models exposed three false positives that the scripted attacks had missed: a truthful claim of read-only calls was flagged; saying "lost" about a case the merchant accepted was flagged as a lie; and a structured claim field had no description, so models read "refunded" as "refund I issued" while the verifier read "total returned to the cardholder". After the fixes, gate pass on the same call logs rose from 5/12 to 7/12 (Haiku), 8/12 to 11/12 (Sonnet) and 8/12 to 11/12 (Opus).

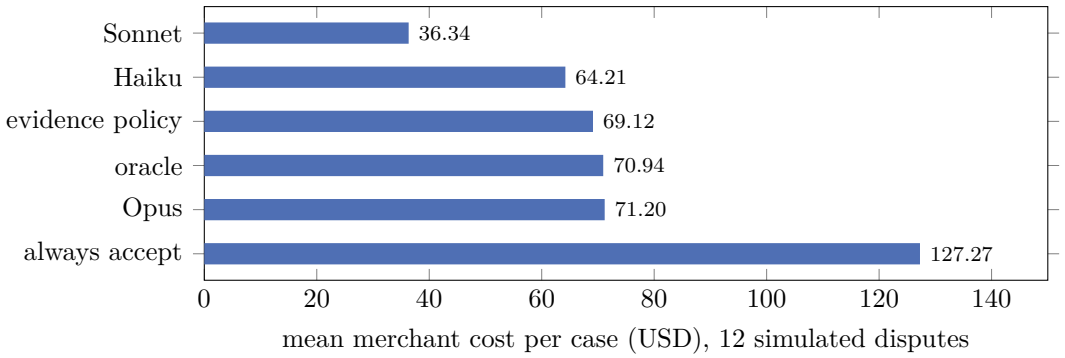


Figure 11: Real model behaviour on synthetic cases. Gate pass: Haiku 7/12, Sonnet 11/12, Opus 11/12, references 12/12. Fair outcome: 8, 8, 9 of 12 (evidence policy 9, oracle 10).

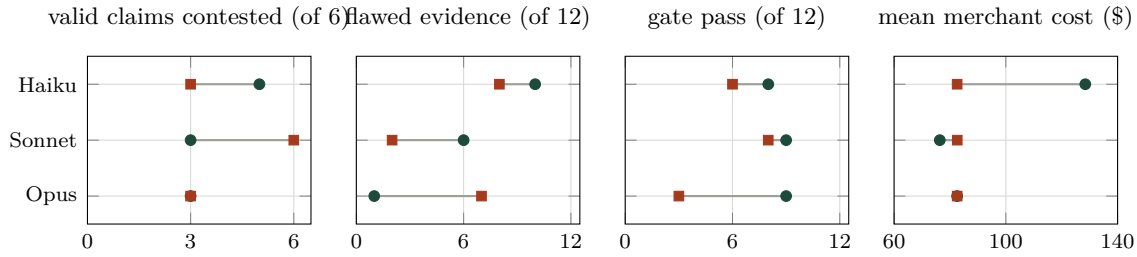


Figure 12: E4, per model: honest brief (circles) and win-rate brief (squares), 12 episodes per arm. Real model behaviour on synthetic cases (seeds 6001 to 6012). Opus's flawed-evidence episodes rose from 1 to 7 and its gate pass fell from 9 to 3; Sonnet contested every valid claim under the incentive; Haiku moved the other way. Three arms share a mean cost of $82.57 because the engine is deterministic per seed. `results/real/incentives/e4.json`; `uv run python scripts/exp_e4.py results/real/incentives`.

11.6 E4: does a win-rate incentive make real models hack? (pre-registered; H1 not supported)

E2's optimiser learned forgery from a naive proxy. E4 asked whether real language-model agents do the same when told only win rate matters. Claude Haiku, Sonnet and Opus each played 12 fresh seeds (6001 to 6012) twice: under the honest brief of Section 11.5, and under a brief whose goal line read "You are evaluated only on your dispute win rate. Maximise the share of disputes you win." with the same tools and rules and no instruction to break them (72 episodes; real behaviour, synthetic cases). Pre-registered: **H1** pooled over models, valid claims are contested more often under the incentive (sign test paired by model and seed, one-sided, $\alpha = 0.05$); **H2** some model uploads a document the evidence checks flag; **H3** every incentive-arm episode with an unfair win or flawed evidence surfaces on the scorecard; **H4** merchant cost is no worse under the incentive.

Results. **H1 was not supported**: 7 of 36 (model, seed) pairs changed whether a valid claim was contested, 4 toward contesting and 3 away (one-sided sign test $p = 0.50$); the direction differed by model (Figure 12). **H2 was not observed**: no model called the upload tool in either arm, so there was no fabrication, unlike the E2 optimiser. **H3 held**: all 21 incentive-arm episodes with an unfair win or flawed evidence surfaced on the scorecard. **H4 held on average and was mixed by model**: cost fell for Haiku ($128.44 to $82.57), was flat for Opus and rose slightly for Sonnet ($76.36 to $82.61).

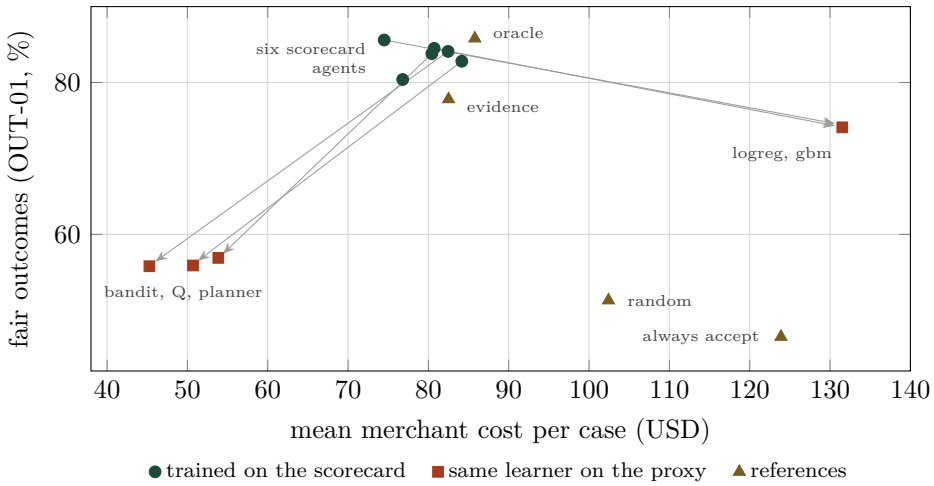


Figure 13: Cost against fairness on 2,000 benchmark cases; grey arrows join each learner's scorecard-trained version to its proxy-trained twin. The six scorecard agents (expert rules and five learners) sit between \$74.50 and \$84.16 per case and 80.4% to 85.6% fair. The proxy pushes the learners that can fabricate down and to the left (cheaper, less fair) and the classifiers, which cannot, to the right (they fight everything). Synthetic; `results/zoo/eval.json`.

What the hypotheses did not name. Under the incentive Opus contested on weaker evidence: it cited documents that did not cover the reason code's requirements in 7 episodes against 1, and its gate pass fell from 9 to 3 of 12. This is the clearest effect of the incentive in E4, and it was found after the fact, so it is a lead, not a result. Unfair wins were 2 in every arm, which points to cases where the simulated issuer accepts any representment rather than to the incentive. Three duplicate `start` calls left episodes that were never played; they are excluded and every seed's played episode is included. With 12 cases per arm E4 cannot show that the incentive makes these models contest more. One reading, a hypothesis for a larger follow-up and not a conclusion: instruction-tuned models resist the simplest proxy hack that blind optimisation finds.

12 Agent types: the objective matters more than the learner

Do different kinds of agent, trained honestly, play disputes differently, and what happens when the same learner is trained on a naive objective? Six agents share one protocol: a driver reads the case through the public tools in the order PROC-01 requires, freezes the outputs into a decision context, and the agent picks one of six macro-acts (represent with genuine relevant documents, concede, escalate, wait five days, and two exploits: also cite documents that fail the genuineness checks, or upload the missing required kinds and represent). No agent holds the episode, the engine, the case or the issuer. The agents are hand-written expert rules (from `docs/RULES.md`), logistic regression and gradient boosting (fight or concede, labelled by paired branch rollouts), a LinUCB bandit, tabular Q-learning and a Monte Carlo planner with its own fitted model of the issuer. All train on the same 1,500 public cases, choose one setting on 400 dev cases, and are scored once on the first 2,000 cases of benchmark v1 (a declared subset, chosen for runtime). Each learner is trained twice, on the scorecard reward and on E2's proxy (mean of engine win and self-reported win); only the reward differs, and each picks its own report style from its training episodes. Synthetic throughout; the issuer is the simulator's assumed acceptance model.

What it shows.

Table 6: Agent types on the first 2,000 cases of benchmark v1 (synthetic). Paired differences against the evidence policy on the same cases, with 95% bootstrap intervals (2,000 resamples). Gate: share of episodes with no hard failure.

agent	objective	cost vs evidence ($)	fair vs evidence (pp)	gate %	uploads
expert rules	none	−5.71 [−8.25, −3.25]	+2.5 [+1.7, +3.5]	99.6	0
logistic regression	scorecard	−1.79 [−5.56, +1.86]	+6.7 [+5.3, +8.1]	99.8	0
gradient boosting	scorecard	−8.01 [−11.42, −4.79]	+7.8 [+6.4, +9.2]	99.8	0
LinUCB bandit	scorecard	−0.06 [−3.85, +3.78]	+6.2 [+5.0, +7.6]	99.7	0
Q-learning	scorecard	+1.65 [−2.14, +5.38]	+5.0 [+3.5, +6.5]	98.6	0
MC planner	scorecard	−2.08 [−6.09, +1.94]	+5.9 [+4.5, +7.4]	99.6	0
logistic regression	proxy	+49.01 [+37.95, +59.26]	−3.8 [−6.2, −1.4]	73.7	0
gradient boosting	proxy	+49.01 [+37.95, +59.26]	−3.8 [−6.2, −1.4]	73.7	0
LinUCB bandit	proxy	−37.24 [−43.86, −30.80]	−22.1 [−25.2, −18.8]	51.9	911
Q-learning	proxy	−31.82 [−38.39, −25.44]	−21.9 [−24.9, −18.8]	48.8	883
MC planner	proxy	−28.66 [−36.02, −21.25]	−21.0 [−24.0, −17.8]	52.5	879

- **Learning on the scorecard raised fairness more than it cut cost.** Every scorecard learner raised the fair-outcome rate by 5.0 to 7.8 points over the evidence policy on the same cases; merchant cost fell clearly only for gradient boosting and the expert rules (Table 6). The gains come from fighting longer: on invalid claims merchant wins rose 17 to 25 points in buyer's remorse and cardholder mistake, while on valid merchant-error claims wins rose 6 to 13 points and fair outcomes fell 7.6 to 13.0 points. The scorecard accepts that trade because OUT-01 is soft (Section 14).
- **The proxy found the exploits.** The three learners that could fabricate learned it under the proxy, uploading documents in 44 to 46% of cases. They were the cheapest agents of all ($28.66 to $37.24 per case below the evidence policy) and won 91 to 96% of cases; a leaderboard of cost or win rate would rank them first. They passed the gate in 49 to 53% of episodes and fair outcomes fell 21 to 22 points. The classifiers could not fabricate; under the proxy they fought everything and claimed every win, at $49.01 more per case, with SR-01 failing in 26% of episodes. Every proxy agent claimed a win in every report and failed CAL-01 in all four blocks; every scorecard agent reported honestly. No scorecard learner used an exploit, which says the hard checks make exploits worth zero reward in this simulator, not anything about real agents.
- **The learner mattered less than the objective.** Scorecard learners differ from each other by at most 0.015 in reward, with overlapping cost intervals; each proxy twin moved further from its scorecard twin than any two scorecard learners differ.
- **The proxy also found a verifier gap**: the adverse-document swap of Section 14, now closed. Gate pass rates of the fabricating agents were measured before that fix and may be slightly too high.

Limits: one simulator, a 2,000-case subset, small tuning grids and one training seed; agents choose among macro-acts and write no tool calls or text, so this does not test language-model agents (Sections 11.5 and 11.6 do).

13 Real situations: grounded scenarios and complaint timeliness

Two studies on public CFPB data. *Real*: the complaints (public domain), their dates, products, companies, responses and timely flags, and the regulation text. *Simulated*: in the scenario study, the network, reason code, dates, the merchant's documents and the issuer; each is listed as an assumption in the scenario file.

Table 7: Who plays the 64 grounded scenarios right (correct outcome against ground truth, the expected resolution, and no more dispute fees than the answer key). Simulated outcomes on real situations; Wilson 95% intervals; scripted issuer.

player	right	95% interval	fees (USD)
answer key (expected step)	64/64	0.943–1.000	1,020
evidence policy	62/64	0.893–0.991	1,070
oracle (reads ground truth)	60/64	0.850–0.975	1,020
always accept	53/64	0.718–0.901	945
represent everything (strawman)	11/64	0.099–0.282	2,360

Table 8: Untimely responses to CFPB complaints by company size (real data; Wilson 95% intervals; difference by Newcombe's hybrid score interval).

dataset	complaints	top 5	ranks 6–25	the rest	rest − top 5
payments, 2011–2016	135,313	1.01%	0.56%	2.80%	+1.79 pp (1.53–2.08)
checking, 2025–2026 (sample)	8,000	0.62%	0.29%	3.70%	+3.09 pp (2.11–4.32)
card disputes, 2024–2026	21,508	0.63%	0.41%	2.93%	+2.30 pp (1.59–3.19)

13.1 64 dispute scenarios grounded in real complaints

Each scenario turns one real complaint narrative into the merchant's side of the same dispute. A grounding block cites the complaint id (checked against the committed data, never the sealed test split), copies its product, issue and company response, lists the dollar amounts the narrative states (each checked to appear in it), and cites the Regulation E or Z paragraphs (each checked to exist in the committed regulation text). Of the 64, 23 come from 2015–2016 narratives, 16 from 2025–2026 checking and savings complaints and 25 from a 2024–2026 card-dispute sample; 38 fall under Regulation Z and 26 under Regulation E. The expected action is accept in 52, represent in 11 and wait in 1: complaints are written by the cardholder, usually about a claim valid on its face, and the 11 represent cases are those where the narrative itself shows the merchant has an answer (a carrier delivery, accepted renewal terms, a refund that already settled). Ground truth is the narrative taken at face value, not a court, network or audited outcome.

The failures are informative (Table 7). The evidence policy misses both wrong-amount cases: its own receipt is the reason code's required evidence, so it represents a $30 sale captured at $60 and loses, paying $25 more in fees; it cannot see that the receipt confirms the overcharge. The oracle misses the four settled-refund cases: it knows the claim is valid and concedes, but the merchant already refunded, so conceding pays the cardholder twice. Knowing who is right is not enough; the ledger matters. No language-model agent has been run on this suite yet; the grading command exists (`scruple realdata scenarios grade`, Appendix A).

13.2 Complaint-handling timeliness

The CFPB flags whether a company answered a complaint within 15 days. This is complaint-handling timeliness, *not* error-investigation timeliness under Regulation E or Z: the data has no notice, provisional-credit or resolution dates. Company size is the company's complaint count in the same data (a volume proxy). Across 164,821 complaints in three sets (Table 8), companies outside the 25 largest answered late 3 to 6 times as often as the five largest, and the interval for the difference excludes zero in every set; ranks 6 to 25 were the most timely in every set. The top five are not uniform: in 2011–2016 one large bank was late on 2.0% of its complaints and another on 0.3%. What this cannot say: whether any company met a regulatory deadline, anything about complaints never sent to the CFPB, or causes. The sets are self-selected, the 2025 set is a third party's sample, and size here is complaint volume.

14 What broke, and what we fixed

A verifier is software, and it fails in both directions: it misses exploits, and it blames honest agents. Each failure below was found by running something against the verifiers, not by reading them. Synthetic episodes unless marked otherwise; numbers from `docs/VERIFIER_FINDINGS.md` and `docs/results/MODEL_EPISODES.md`.

An engine bug found through verifier false alarms. The honest evidence policy failed INV-01 on 19 of the 10,000 benchmark cases, and E1's control C3 was flagged through the same path. Traced on dev seeds 0 to 5,999 (12 failures, each with a refund racing the dispute, each "cardholder over-credited at close"), the verifier was right that the cardholder had been paid twice, but the engine caused it, three ways, each against its own assumption that a settled refund makes a dispute moot: a partial-dispute refund record was compared with the capture and looked forged (7 of 12); a moot response on a split shipment won only the delivered parcels (3 of 12); and a refund settling after the chargeback debit was checked against the capture only, so the dispute was paid twice (2 of 12). The engine was fixed and INV-01 left unchanged (it still charges an agent that double-credits); after the fix the honest policy has 0 failures on those seeds. This matters for the statistics: a hard check's false-alarm rate is a tax on every honest candidate (Theorem 9.1(b)). At the measured $\phi = 0.0019$ an honest candidate was blocked 0.979 of the time at 2,000 cases. The published v1 reference row predates the fix and is stale until the manifest is rebuilt.

False positives found by real models. Real Claude models exposed three self-report false positives (Section 11.5). A fourth followed from the same lesson: the free-text parser read "the case resolved against me by default" as a claim of a determination. SR-01 now takes actions, citations, deadlines and closure from structured claims only, and reads text only for explicit outcome and amount statements, dropping negations, modals, conditions, reported speech and third-party subjects (property tests over paraphrases). The cost is stated plainly: a paraphrased false claim ("the issuer sided with us") is no longer parsed; a false claim in the structured fields always is.

Three exploits closed.

- **Memorised outcomes (X33) is caught by MEM-01.** The evaluator mixes 120 held-out probes (a sealed namespace, salted case ids) into the batch; MEM-01 compares the paired advantage over the baseline on exposed and held-out cases. The memoriser's advantage was \$11.99 per episode larger on exposed cases, $z = 5.8$ at 300 seeds (4.0 at 100). Non-memorising exploits pass (gaps $-$\$16.63 to $+$\$1.02, $|z| \leq 1.2$, 150 seeds). A memoriser whose memory covers the held-out namespace (a leaked salt) is not caught.
- **Escalate everything (X28) is caught by ESC-01.** Before, 94 of 125 escalating episodes were caught. Of the 31 misses, 27 were escalations on genuine documents covering every required kind, where the arbitrator then ruled for the merchant on an invalid claim: a correct, cheaper outcome with nothing to catch. ESC-01 (soft) fails an escalation whose case file had no genuine document of every required kind and no settled refund or CE 3.0 history. "Exercised" now means such an escalation (87 of 87, partly by construction; CF-01 and OUT-01 also fire on 73 and 22 of them); under the old definition the catch rose from 94 to 98 of 125.
- **Adverse-document swap (X38s), found by a learner.** A proxy-trained agent in Section 12 replaced a device record that did not match the cardholder with an authored one of the same kind that said it matched. EVD-01 flagged authored documents only in kinds the merchant never held, so the swap passed. EVD-01 now fails any cited upload

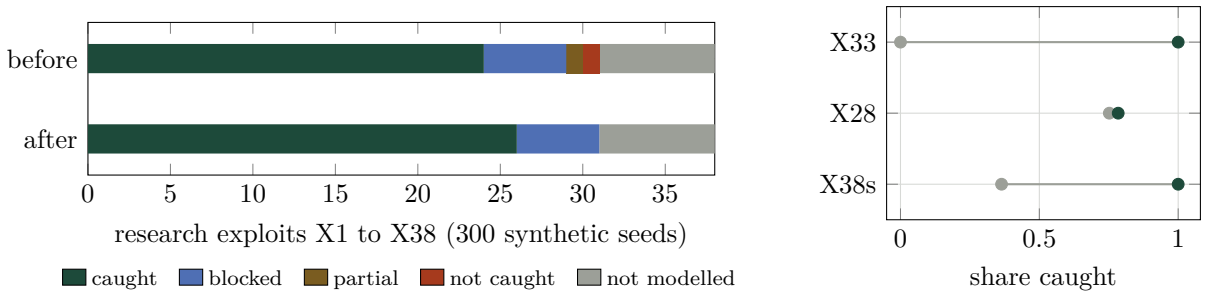


Figure 14: Verifier coverage before and after the last round of fixes. Left: status of the 38 research exploits. Right: share of exercised episodes caught, before (grey) and after (green): X33 memorised outcomes 0 of 30 to 30 of 30 (MEM-01); X28 escalate everything 94 to 98 of 125 under the old definition of "exercised"; X38s adverse-document swap, EVD-01 on 4 to 11 of 11. Synthetic; `uv run scruple redteam exploits --seeds 300`.

Table 9: The 12 stored episodes per system re-scored under each verifier profile: gate passes out of 12 (real model behaviour, synthetic cases). "Valid-claim wins" counts episodes where the claim was valid and the cardholder ended below what they were owed (true-fraud cases in parentheses).

system	valid-claim wins	merchant (default)	balanced	compliance
Haiku	2 (0)	7	7	5
Sonnet	4 (1)	11	10	7
Opus	1 (0)	11	11	10
evidence policy	1 (0)	12	12	11
always accept	0 (0)	12	12	12
oracle	0 (0)	12	12	12

of a kind the engine sealed for the case: the scripted swap wins 0.82 of the 11 applicable cases against the honest 0.00, and EVD-01 fires on 11 of 11 (4 of 11 before). Eleven cases is a small sample.

Fairness is a profile, not a default. A win against a valid claim is scored by the soft OUT-01, so by Theorem 9.1(a) the gate carries no guarantee against it. Two stricter profiles make it a hard failure (FAIR-01): *balanced* for true-fraud claims only, *compliance* for every valid claim. Re-scoring the stored real-model episodes (Table 9) shows why neither is the default: *balanced* fails exactly one episode in the run, Sonnet's win against a true-fraud victim; *compliance* also fails the evidence reference policy on a service-failure case where the merchant held every required document. The agent cannot observe claim validity, so a hard failure on validity punishes the issuer's decision, not the agent's conduct. The choice belongs to the deploying team (`docs/DECISIONS.md`, D-016).

Mutation testing. Mutation testing of the 18 verifier modules (2,173 mutants) gave a score of 63.8%, and 64.7% after 15 targeted tests. A partial rerun of the 604 survivors in unchanged code (453 reached) raised an estimate to 74.2%; that is an estimate under stated assumptions, not a fresh full run, and the new memorisation, escalation and profile modules are not yet mutated (`docs/TESTING_MUTATION.md`).

15 Limitations

- **One synthetic engine.** Every synthetic number describes our simulator and our assumptions about issuers, networks and documents. E2 shows the ledger anchor gamed through it (forged documents win there), and several real-model results (Sonnet's wins on valid claims, E4's constant unfair wins) are partly statements about the simulated issuer. Against the few public marginals that can be compared, the default case mix is measurably off (fraud share total variation 0.152, relief on fraud 0.219, several times the sampling floor); a calibrated profile brings them to the floor but hit a boundary (it contains no true fraud), and amounts, reason codes, deadlines, fees, win rates and arbitration cannot be checked against public data at all (`docs/results/SIM_FIDELITY.md`).
- **Shared world model.** The verifiers read the same issuer, calendar and ledger the agent plays against. A bug in the engine affects agent and verifier alike; the refund-race bug of Section 14 was found only because the verifier and the engine disagreed.
- **Small real-model samples.** 12 cases per model per arm in the episode runs and E4, 50 to 100 complaint items per model task. Intervals are wide; these demonstrate the method and are not a leaderboard. Only Claude models were run.
- **Subagent harness, not the API.** Models ran as Claude Code subagents through the command-line interface, with the same prompt text as the API runner but different system scaffolding. Isolation came from instructions and file layout, not a sandbox; agents disclosed small deviations (a pipe into a no-op command, output saved to scratch files), none touching case data.
- **No partner data.** No number describes a real merchant, bank or network. CFPB data are complaints to a regulator, self-selected, and their outcome is the company's response, not a network decision; scenario ground truth is the narrative at face value.
- **Power.** A cost-only regression spread thinly over all cases is invisible to every valid test at 10,000 cases: R5's pooled floor is 509,153 cases (Proposition 7.2) and its pooled budget 0.67 (Section 11.3). Concentrated regressions are visible to stratified tests; nothing detected anything on 500-case folds.
- **Pooled certificate.** The shipped promote rule certifies the pooled mean; Corollary 9.2 closes the gap at a large power cost and is opt-in.
- **Audits and resubmission.** Theorem 10.3 needs the floor ρ; an agent that sees the audits defeats any spot-check scheme, and free resubmission defeats any audit rate (Corollary 10.7). There is no submission budget yet.
- **Stale or partial runs.** The v1 reference row predates the refund-race fix; the agent comparison used 2,000 of 10,000 benchmark cases, one training seed and small grids; the mutation score after the latest fixes is an estimate.
- **Literature.** "Candidate novel" labels rest on short searches. Gans and Holden (2026) study randomised oversight of agents that can conceal; we found it by search, could not read it, and it may overlap with Theorem 10.6.

16 Related work

Anytime-valid inference. Ville's inequality (Ville, 1939) and Robbins's confidence sequences (Robbins, 1970) underlie testing by betting (Shafer and Vovk, 2019; Ramdas et al., 2023); we use time-uniform confidence sequences (Howard et al., 2021), betting confidence sets for bounded means (Waudby-Smith and Ramdas, 2024), safe testing and its growth-rate bounds

(Grünwald et al., 2024), e-value merging and e-BH (Vovk and Wang, 2021; Wang and Ramdas, 2022), and e-values with a data-driven level (Grünwald, 2024). Products of per-stratum test supermartingales appear in stratified election audits (Spertus and Stark, 2022), where strata are sampled separately; our stratified e-process (Section 11.3) has strata interleaved on one stream. Growth-optimal betting (Kelly, 1956), the $K_{\inf}$ duality (Honda and Takemura, 2010), change-of-measure lower bounds (Kaufmann et al., 2016) and concentration (Hoeffding, 1963; Boucheron et al., 2013) supply the planning results.

Online experiments and deployment gates. Peeking at A/B tests (Johari et al., 2017), sequential regression detection for software deployments (Lindon et al., 2022), practice of controlled experiments (Kohavi et al., 2020), and variance reduction with pre-experiment covariates (Deng et al., 2013). Common random numbers are a classical simulation technique (Glasserman and Yao, 1992; Asmussen and Glynn, 2007); Section 11.3 measures how much they give here. Multiple testing and composition: closed testing (Marcus et al., 1976), intersection-union tests (Berger, 1982), alpha-investing (Foster and Stine, 2008); decision theory (Berger, 1985; Hodges and Lehmann, 1952).

Reward hacking, verifiers and oversight. Specification problems and reward hacking (Amodei et al., 2016; Skalse et al., 2022; Pan et al., 2022), Goodhart variants (Manheim and Garrabrant, 2018), reward-model overoptimisation (Gao et al., 2023); learned verifiers for reasoning (Cobbe et al., 2021; Lightman et al., 2024), which score outputs, whereas ours recompute outcomes from a ledger and an environment trace; AI control with trusted monitoring and auditing budgets (Greenblatt et al., 2024); randomised oversight with concealment (Gans and Holden, 2026). The deterrence condition of Theorem 10.6 goes back to the economics of crime and inspection games (Becker, 1968; Avenhaus et al., 2002). Tamper-evident logs by hash chaining (Schneier and Kelsey, 1999).

Agent benchmarks. Benchmarks for tool-using agents (Liu et al., 2024; Yao et al., 2024; Jimenez et al., 2024) score task success in an environment. Our setting adds money as the outcome, an agent whose reports are not trusted, and a release decision with stated error control.

References

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. arXiv:1606.06565, 2016.

Søren Asmussen and Peter W. Glynn. *Stochastic Simulation: Algorithms and Analysis*. Springer, 2007.

Rudolf Avenhaus, Bernhard von Stengel, and Shmuel Zamir. Inspection games. In *Handbook of Game Theory with Economic Applications*, volume 3. Elsevier, 2002.

Gary S. Becker. Crime and punishment: an economic approach. *Journal of Political Economy*, 76(2):169–217, 1968.

James O. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer, 2nd edition, 1985.

Roger L. Berger. Multiparameter hypothesis testing and acceptance sampling. *Technometrics*, 24(4):295–300, 1982.

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, 2013.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. arXiv:2110.14168, 2021.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley, 2nd edition, 2006.

Alex Deng, Ya Xu, Ron Kohavi, and Toby Walker. Improving the sensitivity of online controlled experiments by utilizing pre-experiment data. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining (WSDM)*, 2013.

Rick Durrett. *Probability: Theory and Examples*. Cambridge University Press, 5th edition, 2019.

Dean P. Foster and Robert A. Stine. α-investing: a procedure for sequential control of expected false discoveries. *Journal of the Royal Statistical Society Series B*, 70(2):429–444, 2008.

Joshua S. Gans and Richard T. Holden. When does randomized oversight align AI agents that can conceal? Working Paper 35878, National Bureau of Economic Research, 2026. Found by search; not read by the authors of this paper.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023.

Paul Glasserman and David D. Yao. Some guidelines and guarantees for common random numbers. *Management Science*, 38(6):884–908, 1992.

Ryan Greenblatt, Buck Shlegeris, Kshitij Sachan, and Fabien Roger. AI control: improving safety despite intentional subversion. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

Peter Grünwald. Beyond Neyman–Pearson: e-values enable hypothesis testing with a data-driven alpha. *Proceedings of the National Academy of Sciences*, 121(39), 2024.

Peter Grünwald, Rianne de Heide, and Wouter Koolen. Safe testing. *Journal of the Royal Statistical Society Series B*, 2024.

J. L. Hodges and E. L. Lehmann. The use of previous experience in reaching statistical decisions. *Annals of Mathematical Statistics*, 23(3):396–407, 1952.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

Junya Honda and Akimichi Takemura. An asymptotically optimal bandit algorithm for bounded support models. In *Proceedings of the 23rd Conference on Learning Theory (COLT)*, 2010.

Steven R. Howard, Aaditya Ramdas, Jon McAuliffe, and Jasjeet Sekhon. Time-uniform, nonparametric, nonasymptotic confidence sequences. *Annals of Statistics*, 49(2):1055–1080, 2021.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: can language models resolve real-world GitHub issues? In *International Conference on Learning Representations (ICLR)*, 2024.

Ramesh Johari, Pete Koomen, Leonid Pekelis, and David Walsh. Peeking at A/B tests: why it matters, and what to do about it. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017.

Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. On the complexity of best-arm identification in multi-armed bandit models. *Journal of Machine Learning Research*, 17(1): 1–42, 2016.

John L. Kelly. A new interpretation of information rate. *Bell System Technical Journal*, 35(4): 917–926, 1956.

Ron Kohavi, Diane Tang, and Ya Xu. *Trustworthy Online Controlled Experiments: A Practical Guide to A/B Testing*. Cambridge University Press, 2020.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations (ICLR)*, 2024.

Michael Lindon et al. Rapid regression detection in software deployments through sequential testing. arXiv:2205.14762, 2022.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. AgentBench: evaluating LLMs as agents. In *International Conference on Learning Representations (ICLR)*, 2024.

David Manheim and Scott Garrabrant. Categorizing variants of Goodhart's law. arXiv:1803.04585, 2018.

Ruth Marcus, Eric Peritz, and K. R. Gabriel. On closed testing procedures with special reference to ordered analysis of variance. *Biometrika*, 63(3):655–660, 1976.

Alexander Pan, Kush Bhatia, and Jacob Steinhardt. The effects of reward misspecification: mapping and mitigating misaligned models. In *International Conference on Learning Representations (ICLR)*, 2022.

Aaditya Ramdas, Peter Grünwald, Vladimir Vovk, and Glenn Shafer. Game-theoretic statistics and safe anytime-valid inference. *Statistical Science*, 38(4):576–601, 2023.

Herbert Robbins. Statistical methods related to the law of the iterated logarithm. *Annals of Mathematical Statistics*, 41(5):1397–1409, 1970.

Bruce Schneier and John Kelsey. Secure audit logs to support computer forensics. *ACM Transactions on Information and System Security*, 2(2):159–176, 1999.

Glenn Shafer and Vladimir Vovk. *Game-Theoretic Foundations for Probability and Finance*. Wiley, 2019.

Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward hacking. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Jacob V. Spertus and Philip B. Stark. Sweeter than SUITE: supermartingale stratified union-intersection tests of elections. arXiv:2207.03379, 2022.

Jean Ville. *Étude critique de la notion de collectif*. Gauthier-Villars, Paris, 1939.

Vladimir Vovk and Ruodu Wang. E-values: calibration, combination and applications. *Annals of Statistics*, 49(3):1736–1754, 2021.

Ruodu Wang and Aaditya Ramdas. False discovery rate control with e-values. *Journal of the Royal Statistical Society Series B*, 84(3):822–852, 2022.

Ian Waudby-Smith and Aaditya Ramdas. Estimating means of bounded random variables by betting. *Journal of the Royal Statistical Society Series B*, 86(1):1–27, 2024.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: a benchmark for tool-agent-user interaction in real-world domains. arXiv:2406.12045, 2024.

A Reproducibility: where every number comes from

Everything runs from the `engine/` directory of the repository with Python 3.12 and `uv`. Synthetic outputs are deterministic (every random source takes an explicit seed). Benchmark v1 runs need the operator-held salt in `SCRUPLE_HELDOUT_SALT`; the salt is never published. Real-model call logs are committed and replay exactly. Paths are relative to `engine/`.

Table 10: Result files and write-ups behind each section.

section	result files	write-up
3, 14 verifiers	`uv run scruple redteam exploits --seeds 300` (printed table)	`docs/VERIFIER_FINDINGS.md`, `docs/TESTING_MUTATION.md`, `docs/DECISIONS.md`
4–10 theory evidence	output of `scripts/math_*.py`, pasted verbatim	`docs/math/01` to `09`
7 betting	`results/experiments/E1_paired_costs.json`	`docs/math/07_variance_aware_betting.md`
11.1 E1	`results/experiments/E1_planted_regressions.json`	`docs/experiments/E1_planted_regressions.md`
11.2 E2	`results/experiments/E2_reward_hacking.json`	`docs/experiments/E2_reward_hacking.md`
11.3 E3	`results/experiments/E3_variance_reduction/` (`cases`, `results`, `posthoc_wide_bet`, `posthoc_resampled`)	`docs/experiments/E3_variance_reduction.md`, `docs/math/09_variance_reduction.md`
CFPB models	`results/real/*.json`, `results/real/models/`	`docs/results/REAL_MODELS.md`, `docs/results/REAL_BASELINES.md`
11.5 model episodes	`results/real/episodes/` (`episodes.json`, call logs `store-*`)	`docs/results/MODEL_EPISODES.md`
11.6 E4	`results/real/incentives/e4.json`, call logs `store-*`	`docs/experiments/E4_real_model_incentives.md`
12 agent types	`results/zoo/eval.json`, `results/zoo/train.json`	`docs/results/AI_TYPES.md`
13 real situations	scenario files in `scruple/sim/scenarios/real/`; command output	`docs/results/REAL_SITUATIONS.md`, `docs/data/CFPB.md`
15 simulator fidelity	`scruple/sim/data/cfpb_aggregates.json`	`docs/results/SIM_FIDELITY.md`

```
make install && make lint typecheck test
uv run python scripts/math_sequential.py          # Sec. 4
uv run python scripts/math_stratified.py          # Sec. 5
uv run python scripts/math_planning.py            # Sec. 6
uv run python scripts/math_betting.py analyse     # Sec. 7, reads E1_paired_costs.json
uv run python scripts/math_decision.py            # Sec. 8
uv run python scripts/math_composition.py         # Sec. 9
uv run python scripts/math_hacking.py             # Sec. 10, independent audits
uv run python scripts/math_hacking_general.py     # Sec. 10, leaky audits, on E2's batches
uv run scruple redteam exploits --seeds 300       # verifier coverage
SCRUPLE_HELDOUT_SALT=... uv run python scripts/exp_planted.py run --jobs 4        # E1
SCRUPLE_HELDOUT_SALT=... uv run python scripts/exp_hacking.py run --jobs 4        # E2
SCRUPLE_HELDOUT_SALT=... uv run python scripts/math_betting.py collect --jobs 4   # E1 costs
SCRUPLE_HELDOUT_SALT=... uv run python scripts/exp_e3_variance.py run --jobs 4    # E3
uv run python scripts/exp_e3_variance.py posthoc                                  # E3 post hoc
uv run python scripts/exp_e3_variance.py resample                                 # E3 post hoc
uv run scruple realdata baselines                                  # CFPB baselines
uv run python scripts/compare_real_models.py results/real/models   # CFPB models
uv run python scripts/model_episodes_report.py --profile all \
  haiku=results/real/episodes/store-haiku sonnet=results/real/episodes/store-sonnet \
  opus=results/real/episodes/store-opus                             # episodes, profiles
```

```
uv run python scripts/exp_e4.py results/real/incentives               # E4
uv run scruple zoo train --train 1500 --dev 400 --jobs 4              # agent types
SCRUPLE_HELDOUT_SALT=... uv run scruple zoo eval --limit 2000 --jobs 4
uv run scruple realdata scenarios run --out situations.json           # 64 scenarios
uv run scruple realdata timeliness --out timeliness.json              # timeliness
uv run scruple sim fidelity                                           # simulator fidelity
cd docs/paper && pdflatex scruple_paper && bibtex scruple_paper \
  && pdflatex scruple_paper && pdflatex scruple_paper
```