FlowX.AI

TECHNICAL PAPER

CASS · WHEN DOES A LEARNED CONSEQUENCE GATE FOR AGENT ACTIONS PAY OFF?

Before the Agent Acts

A small learned gate that scores an agent's proposed action before it executes, and a pre-registered account of where it helped, where simpler guardrails did better, and why it did not pay off in two simulated loan processes.

Bogdan Răduță

Head of Research, FlowX.AI

bogdan.raduta@flowx.ai

FlowX.AI · Applied Research

November 2026 · Version 1.0.0 · doi.org/10.5281/zenodo.24077491

ABSTRACT

An agent in a bank's workflow proposes typed actions whose business outcome is known only later. We ask when a small learned gate pays off: one that predicts before execution whether a proposed action will turn out bad and allows, replans or escalates it. CASS combines published parts: gradient-boosted trees over an explicit Markov state and a typed action, split-conformal thresholds calibrated on good actions, and a nearest-neighbour out-of-distribution flag. On SimBank, a public loan-process simulator, with pre-registered hypotheses and labels that require the simulator's counterfactual replay (privileged supervision absent from an ordinary outcome log), the gate ranks bad candidate actions better than the strongest simple baseline when the log contains a randomized slice (AUROC 0.740 against 0.676) and holds its false-alert targets on later logged actions. It falls short where an agent's gate is needed: its calibration does not transfer to actions the bank did not take (ECE 0.094), a GRU trajectory guardrail stops more bad candidate actions at equal false-alert rate, and a one-line lookup of unseen action types removes more harm on a held-out type than the out-of-distribution flag. On a synthetic stale-fact process with observable labels, adding validity scores from a companion model did not reduce failed submissions (relative reduction −0.004, interval [−0.010, 0.002]), raw signal counts did better, and the gate with validity scores showed no demonstrated cost saving against no gate. In a redesigned follow-up that tracks every case to its end (reported without a verdict for lack of power), a rule that checks an action's facts when its estimated immediate saving, net of check cost, exceeds a tuned threshold cost 1.78 units more per enrolled case than no intervention (interval 0.40 to 3.16; completion 1.6 points higher), and no difference from a tuned risk rule was detected.

KEYWORDS Agent guardrails · Consequence prediction · Conformal prediction · Prescriptive process monitoring · Pre-registration · Negative results

1 Introduction

Language-model agents are being connected to business systems where an action, once executed, has consequences that a bank or insurer records weeks later: a loan is accepted or cancelled, an offer is unprofitable, a case needs rework. A guardrail that sees only the text of an action cannot know these consequences; the institution's own history of outcomes can. This paper studies a *consequence gate*: a small model, owned by the institution and trained on its own workflow outcomes, that scores a proposed agent action before it executes and returns one of `allow`, `replan` or `escalate`, with calibrated probabilities, a false-alert guarantee of stated scope, and an abstain flag for inputs unlike its training data. Which outcomes such a history actually contains matters. A log records the executed action's observable consequences (rejection, rework, cost), not what an alternative would have produced. Our main simulator study labels an action bad relative to its alternatives, which needs the simulator's counterfactual replay even for the actions the bank took; it therefore measures what a gate can learn from privileged supervision, not from an ordinary outcome log.

What is and is not claimed. The techniques are published: outcome prediction for a candidate next action exists in prescriptive process monitoring [1, 35, 45], the three-way gate with split-conformal thresholds on safe examples is DreamGuard's design [27], the class-conditional order-statistic threshold is the Neyman-Pearson umbrella algorithm [42], and nearest-neighbour outlier tests are standard [7, 37]. The contribution we can defend is a combination and its evaluation: a client-owned, per-use-case gate over a typed action and an explicit Markov state, labels defined from business outcomes, conformal false-alert control on good actions together with an out-of-distribution abstain flag, and a pre-registered evaluation with counterfactual ground truth in a simulated bank process. On SimBank the labels are oracle-relative (suboptimality against replayed alternatives) and require counterfactual information at training time, not only at evaluation; on the second, stale-fact process the gate is trained on realised, observable outcome classes. We frame the task as *pre-execution business-outcome prediction for agent actions* and give its metrics (Section 3). We do not contribute a causal method: the S- and T-learners we compare with are established meta-learners [24], and the gate predicts whether an action will turn out bad, not the effect of intervening on it.

The question. We built such a gate from published parts and tested it with pre-registered hypotheses on two simulated loan processes. This paper reports what the tests showed: when a learned pre-action consequence gate pays off, and what stopped it from paying off here. Every hypothesis, metric and threshold was approved before any test data were read (Section 9); amendments, corrections, post-hoc analyses and the full experiment records are in the appendices. Only two experiments measure whether the gate pays off in cost: the stale-fact experiment (E4) and its redesign (Phase 2b-r); the SimBank experiments measure how well the gate flags bad actions.

Findings in brief. Section 6 gives one finding per subsection.

- *What the gate learns* (H1 supported, H5 supported). With labels that need the simulator's counterfactual replay and a randomized slice of 0.1 in the log, the gate ranks bad candidate actions better than the strongest simple baseline. With no randomized logging (fraction 0) it is at chance on candidates and H1 is killed there. The history encodings we tested add less than the registered margin, which does not establish that the state is sufficient.

- *Where calibration holds* (H2 supported on logged actions). The conformal thresholds hold their false-alert targets on the bank's later actions; on candidate actions the bank did not take, which stand in for an agent's proposals, the calibration error is above the pre-registered kill level.
- *Simpler or different guardrails* (H3 killed, H4 supported with a caveat). A recurrent trajectory guardrail in the style of DreamGuard stops more bad candidate actions at equal false-alert rate and is faster than our gate with its flag; the out-of-distribution flag reduces harm on a held-out action type, but a one-line lookup of unseen types removes all of it.
- *Validity scores* (H6 inconclusive). On a second, synthetic loan-intake process with stale facts, where the gate is trained on observable outcome classes, adding validity scores from a companion model did not reduce failed submissions in the primary cell, raw signal counts did better, and the gate with validity scores showed no demonstrated cost saving against no gate.
- *Checks learned from observable labels* (Phase 2b-r, reported without a verdict). A rule that learns the value of a check from a randomized slice of checks cost more per enrolled case than no intervention, with higher completion, and no difference from a tuned risk rule was detected.

H7 (a small model against language models on enterprise side effects) was not run; an offline pilot is in Appendix D. Section 7 states what these results mean for anyone building such a gate.

2 Related work

Pre-execution guardrails for agents. DreamGuard [27] is the closest gate design: a recurrent state-space world model over the interaction history, a three-way PASS, HOLD or BLOCK decision before execution, and two split-conformal thresholds calibrated on safe trajectories, with the threshold formula we also start from. It controls trajectory-level false alerts on safe trajectories, encodes text with a frozen 4B-parameter language model, and reports an average latency of 25 ms per call on one data-centre GPU [27]; its labels come from an agent-safety benchmark, SafetyDrift [14], which models safety trajectories as absorbing Markov chains. ProbGuard [44] learns a discrete-time Markov chain over predicate-abstracted states from traces, estimates the probability of staying safe, and gives a PAC-style sample-complexity analysis. PolicyGuard [46] trains a compact guardrail for policy violations, including from trajectory prefixes. SafePred [9] and TRACES [26] predict risk before or during execution with language-model world models or representation-based auditors; neither calibrates its threshold with a guarantee. Pre-execution consequence prediction also appears for mobile GUI agents [48], embodied agents [32], web agents [33] and network command lines [10]; the last trains a small consequence predictor from execution feedback. EarlyEval [34] uses gradient-boosted classifiers over agent features to predict task outcomes, for evaluation cost rather than gating. A formal alternative checks agent intentions against axioms with an SMT solver, including in a financial-transfer setting [47].

Enterprise world models. World of Workflows [18] reports that frontier language models fail to predict the cascading side effects of actions in an environment built on a commercial IT-service platform, with average audit-prediction overlap and full-action accuracy below 30 percent [18]. Nair et al. [31] report that world models trained offline on enterprise transitions do well in distribution and degrade under deployment shift. Both bear directly on a per-client trained gate: World of Workflows motivates it, and the shift result names the failure mode our later-period and held-out tests are meant to expose.

Predictive and prescriptive process monitoring. Outcome-oriented predictive monitoring predicts a case's outcome from its prefix [40]; the prescriptive line scores candidate next activities by a predicted outcome or indicator [1, 45], fires alarms from thresholded outcome probabilities [16], and estimates intervention effects with causal learners [8, 13]; a review is Kubrak et al. [23]. Shoush and Dumas [35] is the closest prior work in setting: a gradient-boosted outcome model on bank loan logs, a causal effect model and conformal prediction sets that gate interventions. It already combines outcome prediction, conformal confidence and intervention on real loan logs, and it adds what CASS lacks: an estimate of the intervention's effect. CASS differs in what it gates (a proposed agent action, typed, before execution, with an out-of-distribution abstain flag) and in scoring the action's own outcome rather than the effect of intervening; our `replan` and `escalate` are not shown to prevent the outcomes they are triggered by (Section B). SCOPE [13] recommends sequences of interventions by backward induction with causal learners, because interventions on a case interact over time; a gate that scores each action on its own, as ours does, is myopic in exactly that sense. SimBank [12] and ProCause [11] provide counterfactual evaluation for this line; we use SimBank as our instrument.

Calibration, conformal control and deferral. Our thresholds are split conformal [2, 43]; for the false-alert indicator loss, conformal risk control reduces to the same procedure [4], and high-probability

alternatives exist [3, 6, 42]. Weighted conformal prediction addresses covariate shift given a likelihood ratio [41], and conformal off-policy prediction does so under a change of policy [39]. Escalation is a form of deferral [29, 30], and abstention relates to selective classification [17]; we do not learn a rejector jointly with a model of the human.

Logged data and confounding. An outcome model fitted on a bank's log estimates outcomes under the bank's policy. Off-policy evaluation [15, 21, 38], the selective-labels problem [25] and confounding-robust policy learning [22] describe what such a model can and cannot say about actions the bank did not take. Meta-learners such as the S-, T- and X-learner estimate heterogeneous treatment effects from such data with off-the-shelf regressors [24]; we use S- and T-learners only as comparators.

Out-of-distribution detection on tables. The flag is a k-nearest-neighbour distance test [37], compared with isolation forest [28]. On tabular data no unsupervised detector dominates [19], and near out-of-distribution inputs are much harder than far ones [5]. Calibrating the distance threshold on in-distribution data is a conformal outlier test [7].

What we did not find. Among the sources we checked, we did not find a gate for language-model agents whose labels are business outcomes from the deploying institution's own history, with conformal false-alert control on good actions and an out-of-distribution abstain flag, evaluated with counterfactual ground truth in a bank process. This is a narrow combination of established parts; it is not a new causal or conformal method, and the closest work above already covers most of its ingredients. The term "consequence model" is also used for physical asset management [20], in an unrelated sense.

3 Task definition and metrics

Task. At a decision point of a running case, an agent proposes a typed action a (a type and its arguments) in state x. The gate returns a probability that a is bad and a decision: `allow`, `replan` or `escalate`. Whether an action is bad is defined from its business outcome, relative to the alternatives available at the same point. In our SimBank instance (pre-registered, approved before data generation), an action is bad if (a) its case ends in cancellation while at least one alternative at the same point leads to acceptance, or (b) its profit is lower than the best alternative's by more than δ, where δ is 0.05 times the median absolute profit of accepted cases in the training slice, computed once per run on training data only and frozen before later slices were simulated. Under this rule bad actions are common (a mean share of 0.33 of logged and 0.48 of candidate actions on the test slice), so the task is closer to "flag clearly suboptimal actions" than to "flag rare harmful ones".

Populations. We score two populations. *Logged* actions are the ones the bank's policy took. *Candidate* actions are every available action at the factual decision points, scored with the simulator's counterfactual truth; they stand in for an agent's proposals. The candidate population is primary for H1, H3 and H4; H2's false-alert gap is measured on later logged good actions, the scope of the conformal guarantee (Section 4.3). Both are reported for every metric.

Metrics. The false-alert rate (FAR) is the share of good actions that the gate does not allow; the stop rate is the share of bad actions it does not allow. "Stop rate at FAR f" uses the threshold that alerts at most a share f of the good actions in the evaluation set, so scorers are compared at equal FAR; the deployed gate uses the conformal thresholds instead. We also report the area under the stop-rate curve on FAR in [0, 0.2], normalised by the span. Calibration is measured by the expected calibration error (ECE) with 15 equal-mass bins and by the Brier score; ranking by AUROC on each model's raw score. Harm is the regret, in EUR, of bad actions the gate allows (the best alternative's profit minus the action's own); counts are reported too. Latency is wall-clock time per single decision on one CPU thread, and cost per 1,000 decisions is CPU seconds times an assumed CPU price (Section F).

Intervals. Each experiment runs 5 seeds. Intervals are 95 percent percentile intervals of a paired cluster bootstrap (2,000 replicates) that resamples cases within each seed; the statistic is the mean over seeds of the per-seed value. Because the seeds themselves are not resampled, the intervals do not include between-seed variation and are narrow; we show per-seed spreads where they matter.

4 The gate

4.1 Markov formulation

The model assumes that the pair (state, proposed action) is sufficient to predict the outcome: no action history or trace enters as a feature. Where history matters, it is recorded as an explicit state variable. In SimBank the last event row is not Markov, so the state adds the process marking (procedure path, validation phase, status of the headquarters contact), the end times of the two parallel branches, the index of the decision point, and a summary of the repeated noisy quality estimates (their mean and count), next to the case variables (amount, latest quality estimate and its uncertainty, elapsed time, cumulative cost, numbers of contacts and refusals, offered rate and the profitability floor). The latent true quality, every counterfactual outcome, the case number and the seed are excluded. No sensitive attributes exist in SimBank or are used. Whether this state is sufficient is probed by an ablation (E3b), which can show only whether particular history encodings add to it.

4.2 Outcome model and calibration

The bad-action probability comes from gradient-boosted trees (LightGBM, 63 leaves, 600 trees, minimum child size 20, chosen on validation) over the state and typed-action features. Every model is fitted on the logged actions of the training slice. Its target is the frozen rule applied to each logged action, which requires the simulator's counterfactual truth for that action: whether an alternative at the same point would have been accepted, or more profitable. Restricting training to the actions the bank took does not make this target observable. It is privileged supervision. An ordinary outcome log holds the executed action's observable outcome (rejection, rework, cost), not the alternatives' outcomes, so oracle-relative suboptimality is not a quantity a bank records. A retrospective review would not close the gap: reviewers can give judgments or estimates of what an alternative would have produced, a different measurement with its own error, whose agreement with the counterfactual truth would have to be measured; we did not study it (Section 8). The E4 gate (Section B) is trained on realised outcome classes and does not need such labels. Post-hoc calibration is fitted on the logged actions of the later calibration slice. Temperature scaling was chosen on validation among strictly monotone calibrators: isotonic regression had a marginally lower validation Brier score but is a step function whose ties make a conformal threshold coarse. The fitted temperatures range from 1.00 to 1.09 across seeds.

4.3 Conformal gate

Let s be the calibrated probability that a proposed action is bad, and $s_1, \ldots, s_n$ the scores of the good logged actions of the calibration slice, which is later in arrival order than the training slice. For a target false-alert level α the threshold is $\lambda = s_{(k)}$ with $k = \lceil (n+1)(1-\alpha) \rceil$, and the gate alerts when $s > \lambda$. We use two levels: λ_{block} at $\alpha_{\text{block}} = 0.01$ (decision `replan`) and λ_{hold} at $\alpha_{\text{hold}} = 0.05$ (decision `escalate` when $\lambda_{\text{hold}} < s \le \lambda_{\text{block}}$); `escalate` also fires when the out-of-distribution flag is set, and every other action is allowed.

The guarantee, stated precisely. If $n + 1 \ge 1/\alpha$ and a new good action is exchangeable with the good calibration actions, then $\Pr(s_{n+1} > \lambda) \le \lfloor (n+1)\alpha \rfloor / (n+1) \le \alpha$, with the probability taken jointly over the calibration draw and the new action; with continuous scores the left side is at least $\alpha - 1/(n+1)$ [2].

This is false-alert control *on average, only* for good actions exchangeable with the calibration data. It says nothing about how many bad actions are stopped, nothing about good actions proposed by an agent whose distribution differs from the logged ones, and nothing about drift between the calibration period and deployment. Conditional on one calibration set, the false-alert rate is itself random [43]; we therefore also report the Neyman-Pearson threshold [42], which keeps the rate below α with probability at least 1 − 0.05. The formula commonly written as $k = \min(\lceil (n+1)(1-\alpha) \rceil, n)$ breaks the guarantee when $n + 1 < 1/\alpha$, because capping k at n makes the calibration maximum the threshold; our implementation refuses to build a threshold in that case. With both levels and the flag, the share of good actions not allowed is at most α_{hold} plus the flag's rate on good actions (a union bound), not α_{hold} alone. The flag's threshold is calibrated on all in-distribution calibration rows, good and bad together, so its false-out-of-distribution target controls its rate on those rows, not on good actions specifically; the second term is therefore not guaranteed to equal that target, and the sum is not a guarantee on good actions.

4.4 Out-of-distribution flag

The flag is the distance to the k-th nearest neighbour ($k = 50$) in standardized state-action feature space, with up to 20,000 reference rows from the training slice. Its threshold is set on in-distribution calibration rows to a target false-out-of-distribution rate of 0.04, which makes it a conformal outlier test with the same order-statistic argument [7]. The detector (kNN against isolation forest, and k) was chosen on validation at a target of 0.05; the human then moved the target to 0.04 on validation, before the test run, so that H4's pre-registered bound of 0.05 on the false-out-of-distribution rate would not be a coin flip (Section 9). The detector was not re-tuned.

5 Experimental setup

5.1 SimBank (E1, E3, E3b)

Simulator and data. SimBank [12, 36] simulates a bank's loan-application process and can replay a case under every intervention from the same random state. We use three decision points: the procedure (standard or priority), the interest rate (three levels), and whether and when to contact headquarters. Each run has 30,000 cases in arrival order, split into training (cases 0 to 18,000), calibration (to 21,000), validation (to 24,000) and test (to 30,000) slices, each later than the previous; there are no random splits. The logging policy is the bank's own (confounded) policy, mixed per case with a uniformly randomized policy at fractions 0, 0.1, 0.3 and 1.0. Fraction 0.1 is primary; the others are a sensitivity analysis. The test slice has on average 84,857 candidate and 41,046 logged actions per run.

Our wrapper changes SimBank's behaviour in three documented ways: case seeds are derived from the run seed and case index (SimBank accumulates its seed across calls), the customer's acceptance draw gets its own random stream (in SimBank it is a deterministic function of the loan-amount noise), and the two policies are mixed per case so that arrival-order splits are meaningful (SimBank places all bank-policy cases before all randomized ones). With these fixes off, the wrapper reproduces SimBank's own outcome totals exactly. After an action is replayed, the case continues under its own logging policy, so the oracle outcome of a logged action equals its observed outcome.

Comparators. For H1: the prior (global and per typed action), hand-written rules from SimBank's documented bank policy, and logistic regression on the same features; the best of these is chosen on validation per fraction. S- and T-learners, our reimplementation following SCOPE's definitions [13], are reported as causal comparators. SimBank's bundled estimators were not run: they need a library version with no build for our Python version, GPU flags, a hard-coded path and pre-generated external data. For the gate (E3): no gate; the rules gate; a GRU trajectory guardrail in the style of DreamGuard (our reimplementation over the case's event prefix and an embedding of the proposed action, hidden size 64, 12 epochs, chosen on validation; it has no language-model encoder); and a DTMC guardrail in the style of ProbGuard (our reimplementation over documented Boolean predicates, scoring the probability of reaching cancellation in excess of the lowest value among the alternatives, because the bad-action rule is relative). All gates are thresholded by the same conformal procedure. A language-model judge was planned and deferred by the human; it was not built, and H3 is evaluated against the other comparators.

5.2 A stale-fact loan-intake process (E4)

E4 moves to a second simulator: the minimal loan-intake process of the shared simulator `tacit-drift`, built in the companion repository of the validity model (LIV), in which the facts a bank holds about a customer go stale at unknown times. A scripted agent submits a case and later sends its contract using stored facts; outcomes follow rules over the stored and the true values, with costs for rejection, rework and deadline breach that are scenario settings, not claims about real lending. The gate decides at `submit` and `send_contract`: `replan` reverifies the facts the action uses and then executes it, and `escalate` sends the case to a reviewer at a fixed review cost of 15 cost units. Unlike on SimBank, the gate is trained on realised, observable outcome classes, from a warm-up run that reverifies the used facts before a random share of actions. H6 compares the gate with and without

per-fact validity scores from LIV's reference scorer; no gate, and a gate given raw activity-signal counts in place of the scores, are reported. Appendix B gives the full design.

5.3 A redesign with complete follow-up (Phase 2b-r)

Two external reviews found that E4's registered cost measure left out the outcome costs of escalated actions and stopped following cases at the end of the window, that escalation repeated a replan at a higher price, and that the warm-up was not steady. A redesign, written with the companion LIV paper and registered as amendment A1 before any pilot ran, follows every enrolled case to its end, counts every executed outcome, has no escalation and freezes every model before the test. It compares cost-threshold rules, not the conformal gate. A T-learner fitted on a randomized pre-action slice of checks estimates the immediate cost of executing an action as is (r_0) and after checking its used facts (r_1); the *uplift rule* checks iff $r_0 - r_1 - k > \theta_U$, with k the check cost of the used facts, and the *risk rule* checks iff $r_0 > \theta_R$, both thresholds tuned by an inverse-propensity estimate on a calibration window. The labels are observable: the outcomes of the slice's decisions, not a counterfactual replay. Every rule runs under the freshness-only background allocation `fresh_case`. The outcome is total realised cost per enrolled case (every outcome, every check a rule makes and the background checks); completion is the share of enrolled cases that end with a successful contract. The registered contrasts are the uplift rule against no intervention (D2) and against the risk rule (D1), read against the smallest worthwhile saving $X_C = 1$ cost unit per case. The hypothesis, H6r, was moved to reported before the test because the blinded pilot showed it could not reach the planned power (amendment A1b), so it decides nothing; the registered result of this paper remains H6. Appendix C gives the design, the pilot decisions and the remaining reported rows.

6 Findings

Each subsection states one finding and the evidence for it. Pre-registered verdicts are named, reported rows are labelled, and post-hoc analyses are marked. All results come from simulators.

6.1 What the gate learns, and from what

On SimBank the training labels mark an action bad relative to its alternatives, which requires the simulator's counterfactual replay even for the actions the bank took (Section 4). The results of this subsection and the next two therefore measure what a gate can learn from privileged supervision; Sections 6.4 and 6.5 use observable labels.

H1: outcome prediction. Table 1 gives AUROC and Brier scores at the primary fraction. On candidate actions our model's AUROC is 0.740, against 0.676 for the rules scorer, the best simple baseline chosen on validation (gain 0.064, interval [0.061, 0.067]; Brier difference −0.015, interval [−0.016, −0.014]). H1 is supported. Its registered Brier condition also accepted an interval that includes zero; such an interval would only fail to establish a difference, not show that our model is no worse. Here, and at every fraction where H1 is supported, the interval excludes zero in our model's favour, so the condition is met by a difference. The S- and T-learners reach 0.638 and 0.644. They are not information-matched with the classifiers: they fit the observed profit of logged actions (what a log holds) and score a candidate by its predicted regret, while our model and the simple baselines fit the oracle bad-action labels, and the learners' calibration uses those labels too. Their lower AUROC therefore does not show that our model learns better from the same observational evidence; an operational comparison would give every method the same observable outcomes and evaluate the resulting policies. Logistic regression ranks candidates below chance (0.452) although it is the strongest simple baseline on logged actions (0.707): a model fitted on the bank's choices extrapolates poorly to the actions the bank rarely takes.

Table 1. E1 at randomized fraction 0.1 (test slice, mean over 5 seeds). AUROC on raw scores, Brier on calibrated probabilities. The S- and T-learners are our reimplementation, fitted on observed profit rather than on the oracle labels the other rows use, so they are not an information-matched comparison.

model	AUROC, candidates	Brier, candidates	AUROC, logged
ours (LightGBM, temperature)	0.740	0.219	0.886
prior	0.500	0.273	0.500
prior per typed action	0.448	0.345	0.577
rules (H1 anchor)	0.676	0.233	0.594
logistic regression	0.452	0.336	0.707
S-learner	0.638	0.249	0.617
T-learner	0.644	0.247	0.624

The randomized fraction matters (Table 4). At fraction 0 the bank's policy is deterministic given the state, the log has no support for alternative actions, and our model is at chance on candidates (AUROC 0.485; gain over the prior −0.015, interval [−0.018, −0.012]): H1 is killed there. With randomized logging the gain is 0.071 at 0.3 and 0.058 at 1.0.

H5: Markov ablation. We add history to the state in three ways: counts of earlier activities and actions by type (the aggregation encoding of Teinemaa et al. [40]), a GRU summary of the event prefix (hidden size 16, 12 epochs, cross-fitted by case in 3 folds), and both. At the primary fraction, adding both changes candidate AUROC by 0.0041 (interval [0.0033, 0.0050]), whose upper bound is below the pre-registered margin of 0.01: H5 is supported as registered. The counts alone change it by −0.0002 and the GRU summary alone by 0.0020. On logged actions history slightly lowers AUROC (−0.0041, interval [−0.0050, −0.0033]). This is a measured result for these encodings, this model class, this sample and this logging regime: the tested history encodings add less than the registered improvement. It does not establish that the state is sufficient (Markov) for the process; another encoding or model could extract more. At fraction 0 history helps (0.0263, interval [0.0257, 0.0270]), all of it through the GRU summary (0.0270; counts −0.0009). Without randomized logging the bank's alternatives have poor support, and history can then help through confounding, representation or extrapolation; the result does not identify missing state as the cause (Section 8). Figure 3 in Appendix E shows all fractions.

6.2 Calibration holds on the bank's actions, not on candidate actions it did not take

H2: thresholds and calibration on a later period. On later logged good actions the conformal thresholds give an observed FAR of 0.0503 at α_{hold} and 0.0106 at α_{block}; the mean absolute gaps are 0.0025 and 0.0014 (pre-registered limits 0.01 and 0.005). Per seed the FAR at α_{hold} ranges from 0.0471 to 0.0532, on 27,147 to 27,717 good actions. The ECE on later logged actions is 0.016 (limit 0.03). H2 is supported on its pre-registered population. The Neyman-Pearson thresholds give gaps of 0.0029 and 0.0016. Calibration does *not* transfer to candidate actions: there the ECE is 0.094, above the pre-registered kill level of 0.08. Figure 1 shows the two reliability diagrams; on candidates the model under-predicts at low scores and over-predicts at high ones. The guarantee covers the bank's own good actions, not an agent's proposals, and these numbers show that the difference is not academic. The candidate ECE falls as the randomized fraction grows (0.312, 0.094, 0.078, 0.032 at the four fractions).

FIG. 1 · CALIBRATION ON LOGGED AND CANDIDATE ACTIONS

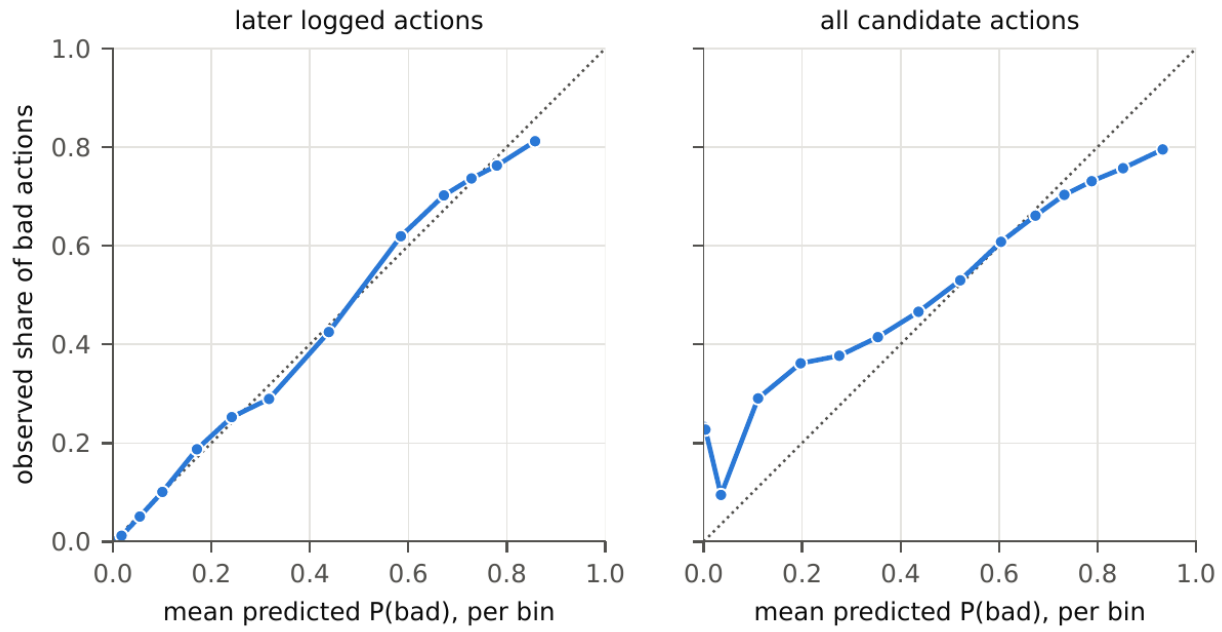


Figure 1. Reliability diagrams of the calibrated bad-action probability, test slice, fraction 0.1, rows of the 5 seeds pooled, 15 equal-mass bins. Left: later logged actions (ECE 0.016, mean over seeds). Right: all candidate actions (ECE 0.094).

6.3 A trajectory guardrail and a type lookup did as well or better

H3: comparison with trajectory and Markov-chain guardrails. At FAR 0.05 on candidate actions our gate stops 0.189 of bad actions (interval [0.185, 0.194]), the GRU trajectory guardrail 0.350 ([0.345, 0.355]), the DTMC guardrail 0.045 and the rules gate 0.000. The difference to the best comparator, the GRU, is −0.161 (interval [−0.165, −0.156]), below the pre-registered non-inferiority margin of −0.02: **H3 is killed.** The GRU is also faster at p95 (0.52 against 0.91 ms for our full gate with the flag) and cheaper. On logged actions the two are level: 0.365 against 0.371, difference −0.006 (interval [−0.015, 0.003]), and the areas under the stop-rate curves are 0.549 and 0.549 (Figure 2). The GRU's lead therefore comes from candidate actions the bank's policy rarely took. We found no leakage (the prefix is a copy of the events before the decision, and the GRU is trained on the same logged rows and labels); our reading, not tested, is that a learned action embedding extrapolates to rare actions better than tree splits do. The lead shrinks as randomized logging grows: the difference is −0.037 at 0, −0.103 at 0.3 and −0.016 at 1.0 (interval [−0.020, −0.012], inconclusive under the same rule). The DTMC guardrail is close to a random gate at this FAR; it is our reimplementation with documented predicates, and a stronger predicate set might do better.

FIG. 2 · STOP RATE AGAINST FALSE-ALERT RATE

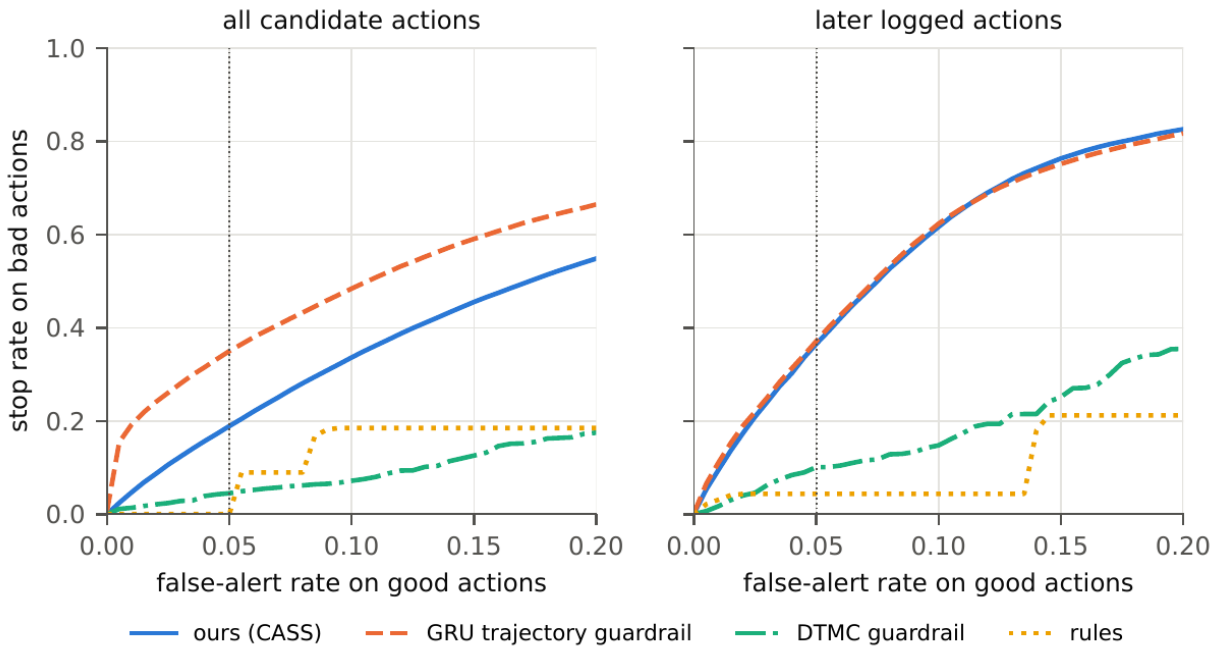


Figure 2. Stop rate on bad actions against false-alert rate on good actions, test slice, fraction 0.1, mean over 5 seeds. The dotted vertical line marks FAR 0.05, where H3 is scored. Each point uses an evaluation-set ranking threshold, the one that alerts the given share of the good actions in the plotted set, so the curves compare rankings; they are not the deployed gate's conformal operating points, whose thresholds are set on the calibration slice. Left: all candidate actions (H3's population). Right: later logged actions.

Latency and cost per decision are in Appendix F.

H4: the out-of-distribution flag. The held-out action type (contacting headquarters) is removed from training and calibration; the model, calibrator, gate and detectors are refitted without it (on average 64,560 held-out candidate rows per run). Adding the flag to the conformal gate reduces harm on these rows by 0.878 (interval [0.876, 0.879]; by count 0.896), at an in-distribution false-out-of-distribution rate of 0.042 on logged rows (bound 0.05). H4 is supported as pre-registered. But the pre-registered lookup baseline, "action type never seen in training", removes all harm on the same rows (reduction 1.0) with no false alarms: **the learned flag is not needed for unseen action types.** On the near-out-of-distribution test (a seen action type with a held-out argument value, the highest interest rate; on average 2,766 rows) the type-only lookup removes nothing (0.0) and the kNN flag removes all harm (1.0). A lookup over type *and* argument also removes all of it there (1.0). The learned flag's value over lookups is therefore shown only against the type-only lookup on near-out-of-distribution inputs; inputs that no lookup can catch, such as unseen regions of the state, were not tested. The flag has a cost: it escalates 22 percent of all candidate actions at the primary fraction (49, 8 and 6 percent at 0, 0.3 and 1.0), because alternatives the bank's policy never takes look out of distribution to a detector fitted on logged rows.

6.4 In the primary cell, validity scores did not reduce the gate's failures

H6. Over the 5 seeds of the primary E4 cell, the gate with validity scores had 17,535 failed submissions and the gate without them 17,466: a relative reduction of −0.004 (interval [−0.010, 0.002]), below the pre-registered 0.10 with an interval that includes zero, so **H6 is inconclusive**. On the subset of

failures that the oracle attributes to stale facts the reduction is 0.008 (interval [−0.005, 0.020]), also not distinguishable from zero.

What changed instead. The validity scores lowered the realised false-alert rate on actions that would have succeeded (0.075 against 0.126). A post-hoc decomposition (D32) shows that the whole drop came from fewer escalations by the out-of-distribution flag (difference −0.064, interval [−0.068, −0.061]), while the false alerts raised by the conformal thresholds rose (difference 0.014, interval [0.012, 0.016]). A gate given raw activity-signal counts in place of the validity scores did better than both deciding arms: it reduced failed submissions by 0.033 against the gate without validity scores (interval [0.026, 0.039]), and the gate with validity scores had a relative reduction of −0.038 against it (interval [−0.044, −0.032]). With its reference settings, the validity model added less to the gate than the raw signals it is built from. In the reported cell with strong signals (C6e), the validity scores did reduce failed submissions, by 0.060 (interval [0.054, 0.067]), below the pre-registered size, and the raw signal counts again did better (Appendix B).

Cost (registered measure). Under the registered cost measure (reported, C6b and C6d), the gate without validity scores cost 0.73 units per case more than no gate (interval [0.53, 0.93]), and the gate with them differed from no gate by −0.13 (interval [−0.34, 0.06]), so no saving is demonstrated.

Cost (post hoc). The registered cost measure left out the outcome costs of escalated actions, which execute. Counting them (D34), the gate with validity scores cost 0.25 cost units per case more than no gate (within-seed bootstrap interval [0.04, 0.43], seed-level t interval [−0.22, 0.71]), so no cost saving against no gate is demonstrated and a higher cost is not established either. The gate without validity scores cost 1.30 more than no gate (seed-level t interval [1.20, 1.40]), and the signal-count gate cost −0.79 ([−0.99, −0.59]), that is, less than no gate. These figures depend on the review cost of 15, which the human had not reviewed before the test run, and are accrued in the test window per case opened in it. A share of 0.51 of the failed submissions of the gate without validity scores are rejections that no gate can prevent (post hoc). Appendix B gives every arm, cell and post-hoc analysis.

6.5 With observable labels, the learned check rule cost more than no intervention

Phase 2b-r replaces SimBank's counterfactual labels with the outcomes of a randomized slice of checks, follows every enrolled case to its end and counts every outcome. The contrasts below are reported, not tested: two-sided 95 percent Student t intervals over 10 seeds of paired per-seed differences, unadjusted and not simultaneous, with exact sign-flip tests beside them.

Table 2. Phase 2b-r primary condition, `fresh_case` allocation: total realised cost per enrolled case (mean over 10 seeds, per-cell 95 percent t interval), its outcome and intervention-check parts per enrolled case (the background spend, equal in every cell, is the remainder), checks per enrolled case, the share of checks whose realised immediate value was not positive (unneeded), the share of decisions where a check would have prevented a failure that went unchecked (missed), and completion. Unneeded and missed use simulator truth and are descriptive. Differences between cells are judged by the paired contrasts in the text. Simulated data.

rule	cost	outcome	checks	checks/case	unneeded (%)	missed (%)	completion (%)
none	35.5 [34.7, 36.2]	30.23	0.00	0.00		100	69.8
age	36.5 [35.0, 38.0]	28.41	2.87	0.56	95	55	71.4
risk	37.2 [36.4, 37.9]	28.57	3.40	0.60	97	66	71.3
uplift	37.2 [35.6, 38.9]	28.37	3.65	0.69	96	55	71.5
always	41.0 [40.2, 41.8]	26.72	9.09	1.77	97	0	72.8

Results (reported, no verdict). Against no intervention (D2), the uplift rule cost 1.78 units more per enrolled case (95 percent interval 0.40 to 3.16; two-sided exact sign-flip $p = 0.006$), a relative cost increase of 5.0 percent (seed bootstrap interval 2.1 to 8.6), with completion 1.6 percentage points higher (1.1 to 2.2; ledger C10b). Against the risk rule (D1), the difference was 0.05 units (−1.25 to 1.36; $p = 0.941$) and completion 0.1 points (−0.6 to 0.8): no difference was detected, and the interval extends beyond X_C on both sides, so neither a worthwhile saving nor its absence is shown (C10c). Against always check, the uplift rule saved 3.79 units (2.46 to 5.13) with completion −1.3 points (−1.9 to −0.7; C10d). The age rule against no intervention differed by 1.05 units (−0.29 to 2.39), and the uplift rule against the age rule by 0.73 (−1.23 to 2.69; C10e). Net of an ongoing slice at the warm-up's rate, which cost 1.81 units per warm-up case, the uplift rule cost 1.88 more than the risk rule (0.57 to 3.19) and 3.61 more than no intervention (2.22 to 4.99; C10f). As registered, this charges the slice to the uplift rule alone, although the risk rule's model and threshold were also fitted on slice decisions, so the net difference against the risk rule overstates the uplift rule's extra cost. In the transfer condition the two contrasts were 0.59 (−0.91 to 2.10) and 1.96 (0.45 to 3.46; C10g).

Table 2 shows where the cost goes. In point estimates every rule that checks lowers the outcome cost per case against no intervention but costs more in checks than it saves; a paired interval establishes the higher total cost only for the uplift rule (D2), the age rule's interval includes zero, and the risk rule and always check have no paired contrast with no intervention. The uplift rule lowers outcome cost from 30.23 to 28.37 and its checks cost 3.65; always check, which misses no preventable failure, lowers it to 26.72 and its checks cost 9.09. Measured with the simulator's truth, between 95 and 97 percent of the checks of every rule had no positive immediate value, so on this measure the learned rules chose their checks no better than always check. In this process, at deployment, a check rarely prevents a failure, and the uplift rule did not find those cases well enough to pay for its checks (D2).

A shift between learning and deployment. The stale rate of the facts used at decisions was 0.094 to 0.108 per seed in the fitting window and 0.018 to 0.022 at the decisions of the `fresh_case` arm without a gate (C10j): the rules learned the value of a check where stale facts were much more common than at those decisions. This is a plausible reason for the many unneeded checks; we did not test it. The interactions with the background allocation, the stationarity check and the pilot decisions are in Appendix C.

7 What this means in practice

> *We built a small consequence gate from published parts and asked when it pays off.*
>
> CASS · FROM THE CONCLUSION

Every statement here comes from simulators and inherits the limits of Section 8; the label in brackets gives the strength of the evidence.

- *With a confounded logging policy, the gate needed randomized logging* (pre-registered, H1; SimBank, privileged labels). With no randomized actions in the log the model was at chance on candidate actions (AUROC 0.485). The useful results depend on a randomized slice that an institution would have to introduce deliberately and pay for.
- *Calibration on past actions did not carry to candidate actions the bank did not take* (pre-registered, H2). The false-alert guarantee covers good actions like the institution's own; on candidate actions, at the primary randomized fraction of 0.1, the calibration error was 0.094, and it fell as the fraction grew. Calibration should be measured on the proposals the gate will see before its thresholds are trusted.
- *Compare with cheap guardrails before building one* (pre-registered, H3 and H4). A lookup of action types never seen in training removed all harm on a held-out type, and a recurrent guardrail over the event sequence stopped more bad candidate actions than the gate.
- *Raw signals before derived scores* (pre-registered null and reported rows, H6). In the primary cell, at reference settings, validity scores did not reduce failed submissions; with strong signals they reduced them below the pre-registered size; in both cells raw activity-signal counts did better.
- *Here, checks rarely changed outcomes, and the learned rule did not pay for its checks* (reported, Phase 2b-r). Between 95 and 97 percent of every rule's checks had no positive immediate value, the uplift rule cost more than no intervention, and the randomized slice needed to learn it cost 1.81 units per warm-up case, more than the smallest worthwhile saving of 1 unit. Before learning which checks to make, it is worth measuring how often a check would change the outcome at all.

These statements would change with evidence from a real workflow: a shadow study that records the pre-action state, the information available at decision time, verification results, eventual outcomes and intervention costs. No such study was run.

8 Limits

Selection bias and counterfactuals. A model fitted on a bank's log estimates the probability of a bad outcome under the bank's policy. It equals the effect of taking the action only without unmeasured confounding and with overlap [22]. Our fraction sweep shows the overlap failure directly: with no randomized logging the model is at chance on candidates. Our SimBank training labels also need counterfactual truth for logged actions, because "bad" is defined relative to alternatives: this is privileged supervision, which SimBank provides and an ordinary outcome log does not. A bank's log records observable rejection, rework and cost; oracle-relative suboptimality is a different target. Retrospective reviewers could supply judgments or estimated alternatives, but not the unexecuted alternative's outcome; such labels would be a different measurement whose agreement with the counterfactual truth would have to be measured before they could stand in for it. A deployable version therefore needs either a target defined on observable outcomes, as in E4, or an estimate of the alternatives' outcomes from data with action support, as the causal learners attempt; our comparison with those learners is not information-matched (Section 6.1). Our metrics measure how well the gate flags actions, scored with the simulator's counterfactual truth; they do not measure the value of the gated policy, because the outcomes after `replan` or `escalate` are not in any real log [15, 25].

Delayed labels. Business outcomes arrive late or not at all. In SimBank, activity durations are fixed constants and every case ends, so we neither faced censoring nor built a time-to-outcome head. A deployment would train only on cases whose outcome has arrived, which biases the training set towards fast outcomes.

Configuration shift. Enterprise dynamics change with configuration and policy versions, and offline-trained world models degrade under such shift [31]. Our later-period test spans one simulator configuration; the held-out tests cover an unseen action type and an unseen argument value, not a changed process. We claim no robustness to configuration shift.

Feedback loop. Once the gate is deployed, the actions it blocks no longer produce outcomes, and the next training set is shaped by the gate itself. Over time this narrows the support of the log, which is the overlap failure above, now caused by the gate. A randomized logging slice for low-impact actions is the only remedy we know of, and it has a business cost.

Gaming. An agent, or a person instructing it, that learns which actions pass the gate can search for actions that score low but are harmful in ways the labels do not capture. The gate sees typed actions and a structured state, which limits but does not remove this. We did not test adversarial proposals.

Average-case calibration. The conformal guarantee holds on average over calibration draws and only for good actions exchangeable with the calibration data (Section 4.3). For one calibration set the realised false-alert rate varies, as the per-seed spread shows; the Neyman-Pearson threshold gives a high-probability version. Neither covers the agent's proposals when they are distributed unlike the bank's actions, where we observed a candidate ECE of 0.094; weighted conformal methods need a likelihood ratio [39, 41]. The bootstrap intervals resample cases within seeds only and are therefore too narrow to describe seed-to-seed variation.

State sufficiency. We did not establish that the state is sufficient. At the primary fraction the history encodings we tested (counts by type, a GRU summary, both) add less than the registered margin, a statement about those encodings, this model class, this sample and this logging regime. At fraction 0

a GRU summary of history adds AUROC. With no randomized logging the alternatives have poor support, and history can help through confounding, representation or extrapolation, so this does not show that a state variable is missing either; the obvious candidate, the history of noisy quality estimates, is already summarised in the state by its mean and count. Whether a state definition suffices is a property of that definition and a process, and must be re-tested for every use case.

Scope of the evidence. The results of E1 to E3b come from one public simulator of one loan process with three decision types, modified by our wrapper in three documented ways. E4 comes from a second, synthetic process built for this study and its companion, with scenario settings (change rates, outcome rules, costs, review cost) that are not estimates of any real process, and with the validity model at its reference settings rather than settings tuned for this process. The bad-action rule makes bad actions common. The comparators are our reimplementations in the style of the cited systems, not those systems; the language-model judge was not built. No client data were used, and no result here was obtained in a regulated deployment.

Experiment-specific limits. The limits specific to E4 (a warm-up that is not steady, pooled thresholds, unavoidable rejections, a weak validity signal under calendar-only logging, escalation equal to replan, an unapproved review cost, calibration rows recorded after the slice's checks, the flag's calibration population, a reused calibration window, truncated follow-up) and to Phase 2b-r (a myopic score transported to deployment, thin support, the slice's cost, frozen models in one scenario, pilot decisions, no verdict, registered quantities not computed) are in Appendix G.

9 Reproducibility

Pre-registration and protocol. The Phase 1 hypotheses (H1–H7), metrics, success and kill thresholds and the bad-action rule were approved by the human before any experiment ran and were not changed. Model choices were tuned on the validation slice only, written into configuration files and frozen by the human before the single test run. At the freeze the human made four decisions: the evaluation populations (candidates for H1, H3 and H4; later logged good actions for H2's false-alert gap), regret as the harm cost, our full gate with the flag for H3's latency condition, and the flag's target false-out-of-distribution rate moved from 0.05 to 0.04 on validation (Section 4). The rules that turn the test summaries into a status for each hypothesis (which fraction decides, populations, intervals, how the best comparator is chosen, taking the stricter reading where a rule was ambiguous) were written and committed before the test run.

Later protocols, corrections and an incident. The E4 protocol, a correction and post-hoc analyses after an adversarial review (D32), corrections after two external reviews (D34), the Phase 2b-r protocol, interpretations of the approved plan, and an incident after the freeze are recorded in Appendix A.

Code, data and environment. SimBank is used from a read-only clone at commit `fd8358bb` (Zenodo DOI `10.5281/zenodo.15574272`, CC BY 4.0); it is not vendored and none of its code is copied. Data are generated by our scripts from seeds; the test runs used code at commit `aec4fa7`, Python 3.13.3 on Apple M5 Max (18 cores, at most six threads during the runs). Every number in this paper is a macro generated from the results files by `scripts/paper_numbers.py`; every figure and the latency table are generated by `scripts/paper_figures.py`; `make paper` rebuilds both and this document. The reliability diagrams read per-row prediction files that are not committed because of their size; the experiment scripts regenerate them deterministically. No language model is used in any experiment.

Archived release. This version (1.0.0) is archived with its code, configurations and results at Zenodo, DOI `10.5281/zenodo.24077491`. It requires the companion LIV release, DOI `10.5281/zenodo.24077472`, which provides the `tacitdrift` simulator and LIV's reference scorer; SimBank is not included and is fetched as described above.

10 Conclusion

We built a small consequence gate from published parts and asked when it pays off. On a public bank-process simulator with pre-registered hypotheses, trained on labels that require the simulator's counterfactual replay of the alternatives (privileged supervision that an ordinary outcome log does not contain), it ranks bad candidate actions better than the strongest simple baseline when the log contains a randomized slice (AUROC gain 0.064), and its conformal thresholds and calibration hold on the bank's later logged actions (FAR gap 0.0025, ECE 0.016), the scope of the guarantee. It falls short where a gate for agents is needed: on candidate actions the bank did not take its calibration does not hold (ECE 0.094), a GRU trajectory guardrail stops more bad candidate actions at FAR 0.05 (0.350 against 0.189) and is faster than our gate with its flag, and a one-line lookup of unseen action types removes all of the harm that its out-of-distribution flag reduces (0.878), while the flag escalates 22 percent of candidates. The history encodings we tested add less than the registered margin (0.0041); this does not establish that the state is sufficient. On a synthetic loan-intake process with stale facts, where the gate is trained on realised outcome classes, adding per-fact validity scores to the gate's state did not reduce failed submissions (−0.004, interval [−0.010, 0.002]; H6 inconclusive), a gate given raw activity-signal counts did better, and counting the outcome costs of escalated actions, which our registered cost measure left out although those actions execute, the gate with validity scores shows no demonstrated cost saving against no gate. In a redesigned follow-up that tracks every enrolled case to its end (Phase 2b-r), whose hypothesis was moved to reported before the test for lack of power, an uplift rule that checks an action's facts when its estimated immediate saving, net of check cost, exceeds a tuned threshold cost 1.78 cost units more per enrolled case than no intervention (interval 0.40 to 3.16; completion 1.6 points higher), and no difference from a tuned risk rule was detected (0.05, interval −1.25 to 1.36); these are reported contrasts, not tested hypotheses. The limits of Section 8 apply to every result above, and the comparison with language models on enterprise side effects (H7) remains to be run.

References

[1] Prerna Agarwal, Avani Gupta, Renuka Sindhgatta, and Sampath Dechu. *Goal-oriented next best activity recommendation using reinforcement learning.* arXiv:2205.03219, 2022.

[2] Anastasios N. Angelopoulos and Stephen Bates. *A gentle introduction to conformal prediction and distribution-free uncertainty quantification.* arXiv:2107.07511, 2021.

[3] Anastasios N. Angelopoulos, Stephen Bates, Emmanuel J. Candes, Michael I. Jordan, and Lihua Lei. *Learn then test: Calibrating predictive algorithms to achieve risk control.* arXiv:2110.01052, 2021.

[4] Anastasios N. Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. *Conformal risk control.* arXiv:2208.02814, 2022.

[5] Mohammad Azizmalayeri, Ameen Abu-Hanna, and Giovanni Cina. *Unmasking the chameleons: A benchmark for out-of-distribution detection in medical tabular data.* arXiv:2309.16220, 2023.

[6] Stephen Bates, Anastasios Angelopoulos, Lihua Lei, Jitendra Malik, and Michael I. Jordan. *Distribution-free, risk-controlling prediction sets.* arXiv:2101.02703, 2021.

[7] Stephen Bates, Emmanuel Candes, Lihua Lei, Yaniv Romano, and Matteo Sesia. *Testing for outliers with conformal p-values.* Annals of Statistics, 51 (1): 149–178, 2023. doi: 10.1214/22-AOS2244.

[8] Zahra Dasht Bozorgi, Marlon Dumas, Marcello La Rosa, Artem Polyvyanyy, Mahmoud Shoush, and Irene Teinemaa. *Learning when to treat business processes: Prescriptive process monitoring with causal inference and reinforcement learning.* arXiv:2303.03572, 2023.

[9] Yurun Chen, Zeyi Liao, Ping Yin, Taotao Xie, Keting Yin, and Shengyu Zhang. *SafePred: A predictive guardrail for computer-using agents via world models.* arXiv:2602.01725, 2026a.

[10] Yuxuan Chen, Rongpeng Li, Zhifeng Zhao, Yuntao Liu, Xing Xu, and Honggang Zhang. *Agentic router: An execution-grounded continual learning approach with memory.* arXiv:2608.09184, 2026b.

[11] Jakob De Moor, Hans Weytjens, and Johannes De Smedt. *ProCause: Generating counterfactual outcomes to evaluate prescriptive process monitoring methods.* arXiv:2509.00797, 2025a.

[12] Jakob De Moor, Hans Weytjens, Johannes De Smedt, and Jochen De Weerdt. *SimBank: from simulation to solution in prescriptive process monitoring.* arXiv:2506.14772, 2025b.

[13] Jakob De Moor, Hans Weytjens, Johannes De Smedt, and Jochen De Weerdt. *SCOPE: Sequential causal optimization of process interventions.* arXiv:2512.17629, 2026.

[14] Aditya Dhodapkar and Farhaan Pishori. *SafetyDrift: Predicting when AI agents cross the line before they actually do.* arXiv:2603.27148, 2026.

[15] Miroslav Dudik, John Langford, and Lihong Li. *Doubly robust policy evaluation and learning.* arXiv:1103.4601, 2011.

[16] Stephan A. Fahrenkrog-Petersen, Niek Tax, Irene Teinemaa, Marlon Dumas, Massimiliano de Leoni, Fabrizio Maria Maggi, and Matthias Weidlich. *Fire now, fire later: Alarm-based systems for prescriptive process monitoring.* arXiv:1905.09568, 2019.

[17] Yonatan Geifman and Ran El-Yaniv. *Selective classification for deep neural networks.* arXiv:1705.08500, 2017.

[18] Lakshya Gupta, Litao Li, Yizhe Liu, Sriram Ganapathi Subramanian, Kaheer Suleman, Zichen Zhang, Haoye Lu, and Sumit Pasupalak. *World of workflows: A benchmark for bringing world models to enterprise systems.* arXiv:2601.22130, 2026.

[19] Songqiao Han, Xiyang Hu, Hailiang Huang, Minqi Jiang, and Yue Zhao. *ADBench: Anomaly detection benchmark.* arXiv:2206.09426, 2022.

[20] Aidan Helmbold. *Consequence models: The missing layer between prediction and decision.* `https://www.sandtech.com/insight/consequence-models`. Accessed 2026-10-05.

[21] Nan Jiang and Lihong Li. *Doubly robust off-policy value evaluation for reinforcement learning.* arXiv:1511.03722, 2015.

[22] Nathan Kallus and Angela Zhou. *Confounding-robust policy improvement.* arXiv:1805.08593, 2018.

[23] Kateryna Kubrak, Fredrik Milani, Alexander Nolte, and Marlon Dumas. *Prescriptive process monitoring: Quo vadis?* arXiv:2112.01769, 2021.

[24] Sören R. Künzel, Jasjeet S. Sekhon, Peter J. Bickel, and Bin Yu. *Meta-learners for estimating heterogeneous treatment effects using machine learning.* arXiv:1706.03461, 2019.

[25] Himabindu Lakkaraju, Jon Kleinberg, Jure Leskovec, Jens Ludwig, and Sendhil Mullainathan. *The selective labels problem: Evaluating algorithmic predictions in the presence of unobservables.* In KDD, 2017. doi: 10.1145/3097983.3098066.

[26] Jiaqian Li, Yanshu Li, Boxuan Zhang, Ruixiang Tang, and Kuan-Hao Huang. *TRACES: Proactive safety auditing for multi-turn LLM agents via trajectory-state modeling.* arXiv:2605.27690, 2026.

[27] Wenhao Lin, Chenyu Yu, Xingwei Lin, Sicong Cao, Xiang Chen, Lei Xue, Le Yu, Letian Sha, and Chunming Wu. *DreamGuard: Efficient runtime guardrail for LLM agents via risk-aware world model.* arXiv:2608.05695, 2026.

[28] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. *Isolation forest.* In ICDM, pages 413–422, 2008.

[29] David Madras, Toniann Pitassi, and Richard Zemel. *Predict responsibly: Improving fairness and accuracy by learning to defer.* arXiv:1711.06664, 2018. NeurIPS 2018.

[30] Hussein Mozannar and David Sontag. *Consistent estimators for learning to defer to an expert.* arXiv:2006.01862, 2020.

[31] Jishnu Sethumadhavan Nair, Patrice Bechard, Rishabh Maheshwary, Surajit Dasgupta, Sravan Ramachandran, Aakash Bhagat, Shruthan Radhakrishna, Pulkit Pattnaik, Johan Obando-Ceron, Shiva Krishna Reddy Malay, Sagar Davasam, Seganrasan Subramanian, Vipul Mittal, Sridhar Krishna Nemala, Christopher Pal, Srinivas Sunkara, and Sai Rajeswar. *Do enterprise systems need learned world models? The importance of context to infer dynamics.* arXiv:2605.12178, 2026.

[32] Massimiliano Pappa, Luca Romani, Valentino Sacco, Alessio Palma, Stéphane Lathuilière, Fabio Galasso, Xavier Alameda-Pineda, and Indro Spinelli. *Describe-then-act: Proactive agent steering via distilled language-action world models.* arXiv:2603.23149, 2026.

[33] Zhouzhou Shen, Xueyu Hu, Xiyun Li, Tianqing Fang, Juncheng Li, and Shengyu Zhang. *World-model-augmented web agents with action correction.* arXiv:2602.15384, 2026.

[34] Yuling Shi, Zhensu Sun, Junsen Dong, Chengcheng Wan, David Lo, and Xiaodong Gu. *EarlyEval: Cheaper agent evaluation via early outcome prediction.* arXiv:2609.02783, 2026.

[35] Mahmoud Shoush and Marlon Dumas. *Intervening with confidence: Conformal prescriptive monitoring of business processes.* arXiv:2212.03710, 2022.

[36] SimBank. *JakobDeMoorKULstudent/SimBank: SimBank: from simulation to solution in prescriptive process monitoring.* Zenodo, software, version v1.0.0, 2025. Creative Commons Attribution 4.0 International.

[37] Yiyou Sun, Yifei Ming, Xiaojin Zhu, and Yixuan Li. *Out-of-distribution detection with deep nearest neighbors.* arXiv:2204.06507, 2022.

[38] Adith Swaminathan and Thorsten Joachims. *Counterfactual risk minimization: Learning from logged bandit feedback.* arXiv:1502.02362, 2015.

[39] Muhammad Faaiz Taufiq, Jean-Francois Ton, Rob Cornish, Yee Whye Teh, and Arnaud Doucet. *Conformal off-policy prediction in contextual bandits.* arXiv:2206.04405, 2022.

[40] Irene Teinemaa, Marlon Dumas, Marcello La Rosa, and Fabrizio Maria Maggi. *Outcome-oriented predictive process monitoring: Review and benchmark.* arXiv:1707.06766, 2017.

[41] Ryan J. Tibshirani, Rina Foygel Barber, Emmanuel J. Candes, and Aaditya Ramdas. *Conformal prediction under covariate shift.* arXiv:1904.06019, 2019.

[42] Xin Tong, Yang Feng, and Jingyi Jessica Li. *Neyman-Pearson classification algorithms and NP receiver operating characteristics.* Science Advances, 4 (2): eaao1659, 2018.

[43] Vladimir Vovk. *Conditional validity of inductive conformal predictors.* arXiv:1209.2673, 2012.

[44] Haoyu Wang, Christopher M. Poskitt, Jiali Wei, and Jun Sun. *ProbGuard: Proactive runtime monitoring for LLM agent safety via probabilistic prediction.* arXiv:2508.00500, 2026. Accepted by the 41st IEEE/ACM International Conference on Automated Software Engineering (ASE 2026). Earlier title: Pro2Guard.

[45] Sven Weinzierl, Sebastian Dunzer, Sandra Zilker, and Martin Matzner. *Prescriptive business process monitoring for recommending next best actions.* arXiv:2008.08693, 2020.

[46] Xiaofei Wen, Wenjie Jacky Mo, Yanan Xie, Peng Qi, and Muhao Chen. *Learning efficient guardrails for compliance.* arXiv:2510.03485, 2026. Accepted by ICML 2026.

[47] Benlong Wu, Weiming Zhang, Kejiang Chen, Han Fang, and Nenghai Yu. *Provably secure agent guardrail.* arXiv:2605.29251, 2026.

[48] Xue Yu, Bo Yuan, Kailin Zhao, Pengshuai Yang, Hong Hu, and Junlan Feng. *SeerGuard: A safety framework for mobile GUI agents via world model prediction.* arXiv:2607.15550, 2026.

Appendix A Protocol details, corrections and an incident

E4 protocol. The H6 rules (deciding cell and arms, the strict reading that counts all failed submissions, equal alphas, the entity bootstrap) were approved by the human before any E4 code existed, and the script that applies them was committed before the test window was run. The E4 configuration was frozen by the main session under a delegation from the human, with a mechanical rule (every validation run completed, background checks and noise draws identical across arms, every metric finite, settings chosen only by the pre-registered tuning rule, nothing changed after validation outputs existed, tests passing, runtime inside the guide); the human had not reviewed the validation outputs or the review cost and delay when the test window was run. Three further disclosures. (i) `gate_with` takes its validity scores from LIV's reference scorer at its default (reference) settings (time-bin width 30, 3 EM iterations), not the per-run settings that LIV's own experiments tune. (ii) The E4 outputs save neither the calibrator chosen per arm nor the out-of-distribution flag per decision, so the flag rates we give are lower bounds (escalations that only the flag can cause). (iii) Arms that receive no signal input are the same simulation in the primary and strong-signal cells, and the signal-count arm equals the gate without validity when signals are off; this is by construction, not a coincidence of the data. The E4 test runs used code at commit `c39dcdc` (companion repository at `14d63a8`), 15 runs of 46 to 59 minutes each with at most 6 threads.

A correction and post-hoc analyses of E4 (D32). After the test run, an adversarial review of the E4 section found a misattribution and several gaps. An earlier draft read the realised false-alert rate above α_{hold} as the conformal guarantee failing to carry to the test window; Section B now separates the conformal part from the out-of-distribution flag, and the abstract and conclusion say which part the validity scores lowered and that their cost advantage depends on the unapproved review cost. The other additions (seed-level intervals, failures no gate can prevent, the ranking of avoidable failures, the escalation accounting, the review-cost dependence and the limits specific to E4) are post hoc and exploratory. They are computed by `scripts/e4_posthoc.py` from the saved E4 outputs, without any new run, and change no verdict. One input is not an E4 output: the reviewer's check of the validity scores on synthetic seeds, whose outputs and script are archived unchanged in the repository and cited only as a synthetic-seed observation.

Corrections after two external reviews (D34). Two external reviews of the research process and of a proposed follow-up found that several sentences of an earlier draft said more than the evidence. No verdict, registered result or frozen configuration changed; the corrections are to wording, with two further post-hoc analyses. (i) The registered E4 cost measure leaves out the outcome costs of escalated decisions; the abstract, introduction and conclusion no longer state a cost saving against no gate, and the post-hoc full accounting (Section B) replaces the earlier break-even costs as the values to read. (ii) The SimBank labels require counterfactual replay (privileged supervision); this now appears in the abstract and the contribution statement, with the retrospective-review suggestion withdrawn. (iii) The causal comparators are not information-matched. (iv) The history ablation is reported as a measured comparison, not as evidence of Markov sufficiency, and a fraction-0 gain is not attributed to missing state. (v) An interval that includes zero is read as no established difference, not as "not worse". (vi) E4 costs are described as accrued in the window per case opened in it, with the incomplete follow-up counted. (vii) The false-alert difference by action kind is described as a plausible

effect of pooled calibration; the flag's calibration population and the reuse of the calibration window are added as limits. (viii) The stop-rate figure's thresholds are labelled as evaluation-set ranking thresholds. (ix) The related work states the overlap with conformal prescriptive monitoring, sequential intervention optimisation and causal meta-learners.

Phase 2b-r protocol. Amendment A1 (with LIV's A2) was written before the pilot; the pilot was blinded to effects; the departures from the registered procedure (amended burn-in rule, slice rate by override, H6r moved to reported, the seed count, a launcher fix, a substitute drift comparison to be reported after the test) were human decisions recorded before the test, and the replacement of that substitute, which failed, by the registered logging-regime check was decided after the test (LIV D70; Section 6.5). The test ran once from one frozen configuration and analysis contract shared with the LIV repository, which runs the harness; a manifest records a cryptographic hash of every input and output file, and this repository's go/no-go script refuses any file that does not match it, any unfrozen input and any pilot seed. The per-case and per-decision files are kept outside version control because of their size; their hashes are in the manifest.

Interpretations disclosed. Where the approved plan left a choice open we took it and record it here: every model is trained on logged actions with labels that require counterfactual truth (privileged supervision that an ordinary outcome log does not hold); calibration and conformal thresholds use logged calibration actions, so the guarantee covers logged good actions; the H1 anchor is chosen per fraction on validation; the DTMC comparator received an "excess" score because its absolute score ranked the relative label below chance on validation; tie-heavy scorers can have empty alert sets at small α, which we report rather than break ties at random; comparator grids were widened when the optimum sat on an edge, which can only strengthen the comparators.

An incident. After the freeze, a unit test that expected the experiment scripts to refuse the test split called the E1 script, which, with the configurations now frozen, started partial test runs. About twenty partial run folders were written. No summary was computed, none was opened or read, and all were deleted unread; the test now uses an unfrozen copy of the configuration. The planned single test run then ran from a clean state. Because the scripts are deterministic given the frozen configurations, the incident could not have influenced any choice or number.

Appendix B E4: stale facts (H6), in full

Setup. E4 moves from SimBank to a second simulator: the minimal loan-intake process of the shared simulator `tacitdrift`, built in the companion repository of the validity model (LIV), in which the facts a bank holds about a customer go stale at unknown times. The world has 5,000 entities; each entity opens a case about once a year. A scripted agent submits the case using the stored employer and income band, and later sends the contract using the stored address, phone and email. Outcomes are decided by rules over the stored and the true values: a wrong employer or an income band stored above the truth gives a rejection, a too-low income band or a wrong address a rework, a wrong phone or email a contact rework, a rework past the case deadline (30 days) a deadline breach, and a true income band at or below a fixed level a rejection whatever the ledger holds (a failure no validity score can prevent). A share of 0.05 of outcomes is replaced by noise. Costs are 100 for a rejection, 30 for a rework, 5 for a contact rework and 50 for a breach. All of these are scenario settings, not claims about real lending. The gate decides at `submit` and `send_contract`: `allow` executes; `replan` reverifies the facts the action uses and then executes; `escalate` sends the case to a reviewer who corrects the used facts, at a fixed review cost of 15 (a delay of 3 days is reported, not simulated). A background calendar refresh with a budget of 1 percent of the total check cost per 7-day epoch is identical in every arm.

The gate is the model of Section 4 on the contract's typed state and action: a multi-class LightGBM over the outcome classes, trained on realised logged outcomes of a warm-up run without a gate that reverifies the used facts before a random share of 0.1 of actions (training from day 91 to 364, calibration to 455, validation to 546), calibrated (none or temperature, chosen on validation), with the conformal alphas of E3 and the kNN flag (k = 50, target false-out-of-distribution rate 0.04). The two deciding arms differ only in the state: `gate_without` sees the facts' values, ages and verification methods; `gate_with` also sees, per fact, the probability that it is still valid, from LIV's reference scorer refitted every 91 days on ledger views only. Two arms are reported and never decide: no gate, and `gate_signals`, which receives a monotone transform of the raw count of activity signals since each fact's last verification instead of the validity probability, to show whether the validity model adds more than its inputs. Every gated arm chooses its LightGBM settings from the same 12-point grid by multi-class Brier score on validation. H6 is decided on the primary cell (moderate heterogeneity, dependencies and signals on); a cell with signals off (where the validity model can use little beyond fact age) and one with strong signals are reported. Each cell runs 5 seeds (100 to 104) over the test window from day 546 to 1,092. The configuration was frozen after the validation runs, and the test window was run once (Section 9).

H6 is evaluated under the strict reading fixed before any E4 run. The statement says failures "caused by stale facts", the metric counts failed submissions, so H6 is decided on *all* failed submissions: every `submit` and `send_contract` decision of a case opened in the test window whose realised class is a rejection, a rework or a deadline breach; escalated decisions count as not failed and pay the review cost. The subset attributed to stale facts by the oracle is reported. Both arms use the same conformal alphas. The statistic is the relative reduction $1 - F_{\text{with}}/F_{\text{without}}$ of failed submissions totalled over seeds, with a paired cluster bootstrap that resamples entities (whose cases share facts) within seeds, 2,000 replicates.

H6. Over the 5 seeds the gate with validity scores has 17,535 failed submissions and the gate without them 17,466: a relative reduction of −0.004 (interval [−0.010, 0.002]). The estimate is below the pre-registered 0.10 and the interval includes zero, so **H6 is inconclusive**: the success rule is not met, and neither is the kill rule (interval upper bound at or below zero). Read plainly, pricing the validity scores into the gate did not reduce failed submissions. On the oracle-attributed stale-caused subset the reduction is 0.008 (interval [−0.005, 0.020]), also not distinguishable from zero.

What the validity scores did change. At the same alphas the gate with validity scores raises fewer false alerts on actions that would have succeeded (realised FAR 0.075 against 0.126, difference −0.050, interval [−0.054, −0.047]) and escalates less (0.069 against 0.107 of decisions), so its expected cost per case is lower by −0.87 cost units (interval [−1.07, −0.68]; 39.95 against 40.81). The cost advantage holds at the review cost of 15, which the human had not reviewed before the test run, and depends on it. These are reported measures, not a pre-registered hypothesis. Two properties of this registered cost measure limit how it may be read. It leaves out the outcome cost of escalated decisions, although an escalated action executes and its outcome is recorded, so it is not the cost the process incurs; the post-hoc full accounting below adds these costs back. And it is the cost accrued in the test window per case opened in the window, not a complete case cost, because the window ends before every opened case has run its course (post hoc, below).

Where the false alerts come from (corrected). Both realised FARs are above $\alpha_{\text{hold}} = 0.05$. An earlier version of this draft read that as the conformal guarantee failing to carry to the test window. That reading was wrong: the realised FAR counts every good action that is not allowed, including escalations by the out-of-distribution flag, whereas the guarantee, and H2 in Phase 1, concern only the share of good actions whose score exceeds λ_{hold}. Splitting the realised FAR into these two disjoint parts, a decomposition computed post hoc from the saved scores and thresholds (D32), gives a conformal part of 0.0415 without validity scores (interval [0.0396, 0.0435]), below α_{hold}, and 0.0554 with them (interval [0.0533, 0.0576]), above it but within the tolerance of 0.01 that H2 allowed; the signal-count gate is at 0.0450 (interval [0.0430, 0.0470]). Most of the excess comes from the flag: on good actions it alone escalates 0.084 without validity scores (interval [0.081, 0.088]) and 0.020 with them (interval [0.018, 0.022]), against a target false-out-of-distribution rate of 0.04 on the calibration window, so even the union bound of Section 4.3 is exceeded without validity scores. The lower realised FAR with validity scores therefore comes entirely from fewer flag escalations (difference −0.064, interval [−0.068, −0.061]), while the conformal part rises (difference 0.014, interval [0.012, 0.016]). The rise is concentrated on `send_contract`: with validity scores the conformal part is 0.038 on `submit` and 0.074 on `send_contract` (difference 0.036, interval [0.032, 0.040]). A plausible explanation is that one pair of thresholds is calibrated on the good actions of both kinds pooled, which controls the pooled rate and permits unequal rates by kind; the observed difference alone does not establish how much of it pooling caused, which per-kind calibration would test. The validation window, available before the freeze, already showed the pattern: realised FAR 0.101 without validity scores (conformal part 0.047, flag alone 0.055) and 0.079 with them (0.054 and 0.026).

Reported arms and cells. The gate itself matters more than the validity scores: the gate without them reduces failed submissions against no gate by 0.099 (interval [0.094, 0.103]), but at a higher expected cost per case than no gate (difference 0.73, interval [0.53, 0.93]), so at a review cost of 15 the review and replan check costs it adds exceed the outcome costs it avoids; with validity scores the cost difference to no gate is −0.13 (interval [−0.34, 0.06]), not distinguishable from zero. These are registered-

measure differences: both depend on the review cost, and both change when the outcome costs of escalated decisions are counted (post-hoc full accounting below). The gate given raw signal counts does better than both deciding arms: a reduction of 0.033 against the gate without validity (interval [0.026, 0.039]), while the gate with validity scores has a relative reduction of −0.038 against it (interval [−0.044, −0.032]), that is, more failed submissions. With its reference settings, the validity model therefore adds less to the gate than the raw signals it is built from. In the reported cells, the same rule gives the following, never deciding. With signals off, where the validity model can use little beyond fact age, the gate with validity scores has slightly more failed submissions than the gate without them (reduction −0.008, interval [−0.014, −0.002]; the rule would read killed), with a lower expected cost per case (difference −0.43, interval [−0.64, −0.23]). With strong signals the validity scores reduce failed submissions by 0.060 (interval [0.054, 0.067]), below the pre-registered size (the rule would read inconclusive), and the stale-caused subset by 0.165 (interval [0.150, 0.179]), at a lower expected cost (difference −2.34, interval [−2.57, −2.12]). There too the raw signal counts do better: the gate given them reduces failed submissions by 0.146 against the gate without validity (interval [0.139, 0.154]), and the gate with validity scores has a relative reduction of −0.101 against it (interval [−0.108, −0.093]). Table 3 gives every arm.

Post-hoc analyses (not pre-registered). The analyses in this paragraph and the following ones were chosen after the E4 test results had been read, following an adversarial review of this section (decision D32) and, for the last two, two external reviews (D34). They are computed from the saved E4 decisions by `scripts/e4_posthoc.py` (the last two by `scripts/audit_posthoc.py`), with the same entity bootstrap where an interval is given; they are exploratory, decide nothing and change no verdict (H6 stays inconclusive). *Seed variation.* The bootstrap intervals above resample entities within seeds and leave out between-seed variation (Section 3). Per seed, the relative reduction of failed submissions with validity scores is 0.014, −0.010, −0.001, −0.010, −0.013 (seeds 100 to 104); a *t* interval over the 5 seeds is [−0.018, 0.010], against the bootstrap interval [−0.010, 0.002], and for the expected-cost difference it is [−1.36, −0.38] against [−1.07, −0.68]. Neither reading changes: the reduction is not distinguishable from zero and the cost difference at the review cost as run is below zero.

Post hoc: failures no gate can prevent. A true income band at or below the process threshold gives a rejection whatever the ledger holds. 9,033 decisions of the primary cell are of this kind, and they account for 8,855 of the 17,466 failed submissions of the gate without validity scores (a share of 0.51); noise accounts for 1,839 more, and the stale-caused subset is the rest. A replan or an escalation cannot change these rejections, yet the gate with validity scores intervenes on 8,701 of the 9,033 unavoidable decisions, which make up a share of 0.63 of its 13,828 interventions (without validity scores 0.54). Those interventions cost 1.41 per case with validity scores and 1.42 without, more than the premium of 0.73 per case that the gate without validity scores costs over no gate. The H6 statistic, which counts every failed submission as registered, is diluted by these failures in both arms.

Post hoc: ranking of avoidable failures. Restricted to decisions that are neither unavoidable rejections nor touched by noise, the gate's score ranks avoidable bad actions (those whose class, executed as proposed, would not be ok) against good ones with an AUROC of 0.635 without validity scores (interval [0.628, 0.641]), 0.665 with them ([0.659, 0.672]) and 0.717 with signal counts ([0.710, 0.724]), means over seeds. At a matched FAR of 0.05 (the threshold set on each seed's good actions, as in Phase 1) the gate stops 0.125, 0.147 and 0.232 of these actions (intervals [0.117, 0.133], [0.139, 0.156] and [0.221, 0.243]). So the validity scores do improve the ranking of avoidable failures a little (AUROC difference 0.030, interval [0.024, 0.036]; stop-rate difference 0.021, interval [0.013, 0.031]), less than

the signal counts do (0.082 and 0.106), but at the deployed thresholds this did not turn into fewer failed submissions, nor into fewer stale-caused ones (H6 above).

Post hoc: escalation and failed submissions. In this process an escalation does what a replan does (the reviewer reverifies the facts the action uses, then the action executes), at the review cost instead of the check cost, and the registered accounting counts an escalated decision as not failed and leaves its outcome cost out even when the case still fails (rule 6c). The review neither defers nor blocks the action, so it does not prevent the outcome. Counting escalated decisions whose realised class is a failure (305 of 6,872 escalations without validity scores, 201 of 4,396 with them) gives 17,736 failed submissions with validity scores against 17,771 without, a relative reduction of 0.002 (interval [−0.003, 0.007]): the H6 reading does not change. The review delay of 3 days, reported and not simulated, would pass the case deadline for at most 3 escalations in any arm and cell (3 of 6,872 for the gate without validity scores in the primary cell); whether a simulated delay would change the paths is not known.

Post hoc: full accounting of escalated outcomes (D34). This analysis corrects how the registered cost comparisons above may be read; it was added after two external reviews and is computed by `scripts/audit_posthoc.py` from the saved E4 outputs. We add back the realised outcome costs of escalated decisions (21,040 cost units over the 5 seeds without validity scores, 14,070 with them, 15,220 with signal counts), keep the review cost and the replan checks, do not add the reviewer's checks again (the review cost covers them), and leave the common background checks out, as before. The cost per case is then 40.08 for no gate, 41.38 without validity scores, 40.33 with them and 39.29 with signal counts (registered measure: 40.08, 40.81, 39.95 and 38.88). We give each paired difference with two intervals: the go/no-go's entity bootstrap within seeds, and a t interval over the 5 per-seed differences, which includes between-seed variation. With minus without validity scores: −1.05 (bootstrap [−1.24, −0.87]; t [−1.51, −0.59]). Without validity scores minus no gate: 1.30 ([1.10, 1.49]; [1.20, 1.40]). With validity scores minus no gate: 0.25 ([0.04, 0.43]; [−0.22, 0.71]). Signal counts minus no gate: −0.79 ([−1.01, −0.57]; [−0.99, −0.59]). Under full accounting, therefore, the gate with validity scores shows no demonstrated cost saving against no gate: its point estimate is a cost increase, the within-seed interval excludes zero but the seed-level interval does not, so a higher cost across simulated worlds is not established either. It remains cheaper than the gate without validity scores, the gate without them costs more than no gate, and the signal-count gate costs less than no gate. The review cost at which each gate breaks even with no gate falls to 7.95 without validity scores (interval [6.89, 9.01]), 12.91 with them ([11.28, 14.65]) and 21.60 with signal counts ([19.75, 23.37]); below these values the gate is cheaper than no gate. These replace, as the values to read, the break-even costs of the registered accounting (11.0, 16.1 and 25.0), which treated escalated outcomes as free. Between the deciding arms, the full-accounting difference (with minus without validity scores) is −0.06 at a review cost of 0 (interval [−0.25, 0.14], not distinguishable from zero), −0.39 at 5 ([−0.58, −0.20]) and −2.05 at 30 ([−2.26, −1.85]): the advantage comes from the reviews the gate with validity scores saves. In the reported cells, the gate with validity scores minus no gate is 0.44 with signals off ([0.24, 0.64]; t [−0.11, 0.99]) and −1.13 with strong signals ([−1.37, −0.88]; t [−1.96, −0.29]). These figures add costs to the paths the arms actually took; a review that delays the action or changes the case's path cannot be reconstructed this way and would need a new, separately specified experiment.

Post hoc: incomplete follow-up (D34). Every E4 count and cost is accrued in the test window, which ends at day 1,092, for the cases opened in it. A case's initial submission comes 14 days after opening and its deadline 30 days after opening, so the window ends before the last opened cases have run

their course. In each primary-cell arm, 969 of the 37,315 cases (a share of 0.026) have no judged decision at all, and 2,102 (0.056) have a deadline after the window end. A shared cut-off does not by itself make the comparisons unbiased, because the arms change retries, reworks and closures. As a descriptive check, restricted to the 35,213 cases whose deadline falls inside the window, the full-accounting differences are −1.10 for with minus without validity scores ([−1.29, −0.91]; *t* [−1.56, −0.63]) and 0.27 for with validity scores minus no gate ([0.05, 0.47]; *t* [−0.18, 0.72]), close to those above. This is not a complete follow-up; costs in this paper are costs accrued in the window per case opened in the window.

Table 3. E4, test window, totals over the 5 seeds. Failed: failed submissions (all classes); stale: the oracle-attributed stale-caused subset; FAR: realised false-alert rate on actions that would have succeeded, counting escalations by the out-of-distribution flag; conf.: its conformal part, the share of those actions with a score above λ_{hold} (post hoc, D32); esc.: escalation rate; cost: the registered expected cost per case (outcome cost of non-escalated decisions, replan check cost and review cost of 15 per escalation, cost-table units), accrued in the window per case opened in it; it leaves out the outcome cost of escalated decisions (see the post-hoc full accounting in the text). Only the primary cell's `gate_with` against `gate_without` decides H6. Arms that see no signal input are the same simulation in the primary and strong-signal cells, and `gate_signals` equals `gate_without` when signals are off (identical by construction). Simulated data.

cell	arm	failed	stale	FAR	conf.	esc.	cost
primary	no gate	19,378	8,377	0.000	n/a	0.000	40.08
	`gate_without`	17,466	6,772	0.126	0.042	0.107	40.81
	`gate_with`	17,535	6,720	0.075	0.055	0.069	39.95
	`gate_signals`	16,892	6,097	0.065	0.045	0.070	38.88
signals off	no gate	20,828	9,820	0.000	n/a	0.000	42.01
	`gate_without`	18,623	7,957	0.129	0.046	0.111	42.42
	`gate_with`	18,765	8,009	0.097	0.067	0.084	41.99
	`gate_signals`	18,623	7,957	0.129	0.046	0.111	42.42
signals strong	no gate	19,378	8,377	0.000	n/a	0.000	40.08
	`gate_without`	17,466	6,772	0.126	0.042	0.107	40.81
	`gate_with`	16,415	5,655	0.059	0.049	0.068	38.48
	`gate_signals`	14,912	4,217	0.058	0.046	0.081	36.89

Appendix C Phase 2b-r: design, pilot decisions and further reported rows

Why a second experiment. E4 left H6 inconclusive, and two external reviews then found that its cost measure left out the outcome costs of escalated actions and stopped following cases at the end of the window, that escalation repeated a replan at a higher price, and that the warm-up was not steady (Section 8). We wrote a redesign with the companion LIV paper and registered it as amendment A1 before any pilot ran: one harness shared by both papers, fresh seeds, every enrolled case followed to its end, every executed outcome counted, no escalation, and every model frozen before the test. H6, its verdict and every E4 number above are unchanged. The design was written after the Phase 2 results and the reviews were known.

The question, and its status. H6r asks whether, under the freshness-only background allocation `fresh_case`, an *uplift rule* (check the facts an action uses if the estimated immediate cost saved by checking them, net of their check cost, exceeds a tuned threshold) lowers total realised cost per enrolled case against both a tuned *risk rule* and no intervention, without lowering case completion by more than $m = 1$ percentage point against either. It was registered as this paper's deciding hypothesis. The blinded pilot showed that it could not reach the planned power with the seeds available (below), and before the test the human moved it to reported (amendment A1b). H6r therefore decides nothing: we report its components with intervals and without a verdict, and the registered result of this paper remains H6.

Design. The process is the loan-intake process of E4, with the scenario settings, cost table and primary cell of the companion LIV experiment, 5,000 entities per world and every rework priced at 30 except a contact rework at 5 in every model, rule and evaluation. Before the test the organisation logs uniformly random background checks (1 percent of total fact check cost per week) and a randomised pre-action slice that checks an action's used facts right before it executes with probability 20 percent, recording the state before the check. After a burn-in of 2184 days the models are fitted on the next 364 days, their settings selected on the following 91 days and their calibration and thresholds set on the 91 days after that; everything is then frozen. Every arm runs on its own fork of the shared warm-up with common random numbers; the cohort is every case opened in the 546-day enrolment window (days 2730 to 3276), followed 30 days beyond it so that every enrolled case ends. The primary outcome is total realised cost per enrolled case: every realised outcome of every action, every check a decision rule spent, and the background spend of the enrolment window per enrolled case. Completion is the share of enrolled cases that end with a successful contract.

Both learned rules share a T-learner: m_0 is fitted on the slice's untreated decisions and m_1 on its treated ones, both multi-class LightGBM models over the outcome classes with the encoder of Section 4 and without validity scores; r_0 and r_1 are their expected immediate costs of executing the action as is and after checking its used facts. The uplift rule checks iff $r_0 - r_1 - k > \theta_U$, with k the check cost of the used facts; the risk rule checks iff $r_0 > \theta_R$. Both thresholds are chosen on the calibration window, among the thresholds at intervention rates in steps of 5 percent, by an inverse-propensity estimate of immediate cost (outcome plus check) over its randomised decisions. The tuned uplift threshold varied widely across seeds: it was $-\infty$ on seed 200, where the rule therefore checks every decision, negative on 4 of 10 seeds counting that one, and at most 12.1; a negative threshold checks

actions whose estimated saving is below their check cost. A check examines every used fact and the action then executes at once; there is no escalation, no conformal threshold, and the out-of-distribution flag is computed but acts in no arm. H6r therefore concerns cost-threshold rules, not the conformal gate of Section 4: a negative result here does not refute H2 or the Phase 2 gate results, and a positive one would not have validated them. The uplift score estimates an immediate effect of checking, so the rule is myopic by construction. Reported cells, without verdicts, are always check, an age rule under `fresh_case` only (check if the oldest used fact is older than a threshold tuned the same way), the no-intervention, risk, uplift and always-check rules under LIV's avoidability-aware allocation `conseq_avoid`, and a transfer condition whose warm-up logging is a calendar review round robin with a slice of 5 percent. Intervals are two-sided 95 percent Student t intervals over 10 seeds (200 to 209) of paired per-seed differences, unadjusted and not simultaneous; exact sign-flip tests over seeds are reported beside them.

Pilot decisions, disclosed. A blinded pilot on 8 seeds that are never confirmatory (300 to 307) wrote only standard deviations, correlations, support tables, harness checks and timings. (i) *Power.* With the pilot's variances, H6r would be supported with probability 0.8 percent at 10 seeds and at most 31 percent at 30 seeds when each true saving equals the smallest worthwhile saving $X_C = 1$ unit, against a target of 80 percent (stop S3). The human moved H6r to reported; the seed count 10 is the one the LIV hypothesis needed. (ii) *Slice support.* The rules need randomly treated decisions where a check could prevent a failure; the registered rule asked for 100 of them per action kind in the fitting window of every pilot seed. At a slice of 10 percent the smallest counts were 85 (submit) and 43 (contract), at 20 percent 161 and 99 (stop S2). The human accepted 20 percent as an override rather than raise the slice, whose cost the uplift rule's saving must cover; on the confirmatory seeds the contract count was again below 100 on 2 of 10 primary seeds. The pilot's slice cost 1.81 cost units per warm-up case, above X_C, which was reported to the human before the freeze. (iii) *Burn-in.* The registered drift rule stopped the pilot (no candidate burn-in passed on every seed); it was amended before any arm ran, with outcome-share gaps judged against 3.36 standard errors (Bonferroni over 64 comparisons), and chose 2184 days. (iv) A launcher bug was fixed before the freeze. The drift item of the freeze checklist could not be computed before the test without using the test window, so before the test the human chose a substitute comparison to report after it (LIV D66); after the test that substitute failed on every seed for a reason unrelated to drift, and the human replaced it with the registered logging-regime check (LIV D70, below). The burn-in and support rules read simulator truth; the burn-in rule was amended after the pilot and the slice rate was set by override. No pilot effect informed any setting, and no truth informed the rules, features, hyperparameters, comparators, X_C, m or the readings. The test ran once.

The registered interaction between allocation and decision rule, $I = C[A,P] - C[A,N] - C[F,P] + C[F,N]$, is 0.79 (0.20 to 1.38) for the uplift rule, 0.80 (0.45 to 1.14) for the risk rule and 1.55 (1.33 to 1.77) for always check (C10h). The last equals the saving of `conseq_avoid` over `fresh_case` without a gate by construction, because the two allocations cost the same under always check. We report the estimates and their intervals only.

Stationarity and deployment shift, reported after the test. The logging regime, continued alone through a test-length window on each confirmatory seed, was within the amended drift rule on 10 of 10 seeds in the primary condition and 10 of 10 in the transfer condition (C10i). Deployment was a different regime: the stale rate of the facts used at decisions was 0.094 to 0.108 per seed in the fitting

window and 0.018 to 0.022 at the decisions of the `fresh_case` arm without a gate (C10j), consistent with that allocation spending its check budget on the facts of open cases; other arms and the calibration window were not compared. The rules learned the value of a check where stale facts were much more common than at those decisions. This shift is a plausible reason for the many unneeded checks; we did not test it.

Appendix D The World of Workflows offline pilot (E2, not pre-registered)

This subsection reports a pilot. It is not a pre-registered test, it decides no hypothesis, and H7 stays not run. We used only the trajectories shipped with World of Workflows [18] (MIT licence): 403 steps in 67 trajectories, and 1,132 constraint-violation steps in 50 trajectories, with no instance, account, network access or language-model call. The constraint steps repeat only 93 distinct actions (tool and parameters), of which 11 are labelled as violating; the whole pool of 1,444 rows holds 399 distinct actions and 55 tools. Statistical power is set by these distinct counts, not by the row count. Features are the typed action, the step index, the previous tool and flags for tables touched by earlier steps; no pre-action state is available offline. Two derived tasks: (a) which of 23 tables an action modifies (multi-label, macro-F1) and (b) whether it violates a constraint (binary). Models: our LightGBM, the prior, a per-tool lookup and logistic regression, tuned on validation.

A disclosed deviation: grouped, not temporal, split. The shipped data have no time order across trajectories, and a split by trajectory would leak repeated actions. We therefore split by distinct action into 60, 20 and 20 percent groups (training, validation, test), one partition per seed (5 seeds). This deviates from our rule of temporal or arrival-order splits for headline numbers; the pilot is not a headline result. Intervals are 95 percent percentile intervals over 1,000 bootstrap draws of groups, pooled over seeds.

Results. On task (a) (399 groups) macro-F1 is 0.176 [0.163, 0.207] for the prior, 0.708 [0.637, 0.751] for the per-tool lookup, 0.639 [0.585, 0.693] for logistic regression and 0.626 [0.573, 0.683] for our model. The lookup has the highest estimate and our model does not beat the baselines: against the best simple baseline chosen on validation (logistic) the paired difference is −0.014 [−0.050, 0.024]. Which tables an action touches is largely a function of the tool. On task (b) (93 groups, 11 violating) no model is distinguishable from chance: AUROC is 0.655 [0.493, 0.816] for the lookup, 0.571 [0.397, 0.807] for logistic regression and 0.548 [0.374, 0.694] for ours, and every interval contains chance. Violations depend on state that the shipped files do not carry, so this is a data limit as much as a model finding. These are World of Workflows' own evaluation data, so we make no comparison with its published numbers.

Recommendation. We do not set up an instance of that platform now. A full E2 would add pre-action state, many more distinct actions, a real generation order for a temporal split, and the only route to H7; it should be revisited together with the human's decision on language-model baselines and their spend cap. H7, which compares a small model with frontier language models used zero-shot, remains not run.

Appendix E Summary across randomised fractions

Table 4. Sensitivity to the randomized fraction (test slice). Only fraction 0.1 decides; the others are evaluated with the same rules and reported. H1: AUROC gain over the anchor chosen on validation per fraction. H2: candidate ECE (H2 itself is decided on logged actions and is supported at every fraction). H3: stop-rate difference to the GRU at FAR 0.05. H4: share of candidates escalated by the flag (percent). H5: AUROC difference with minus without history.

fraction	H1 gain	H2 ECE, cand.	H3 difference	H4 escalated	H5 difference
0	−0.015 (killed)	0.312	−0.037 (killed)	49	0.0263 (killed)
0.1	0.064 (supported)	0.094	−0.161 (killed)	22	0.0041 (supported)
0.3	0.071 (supported)	0.078	−0.103 (killed)	8	0.0053 (supported)
1.0	0.058 (supported)	0.032	−0.016 (inconclusive)	6	0.0024 (supported)

The negative results are as important as the positive ones. (i) H3 is killed: on the agent's candidate actions, a recurrent trajectory guardrail stops substantially more bad actions than our gate at the same false-alert rate and is faster than our gate with its flag; the combination we study does not improve on that comparator in this setting, and the Markov design is not what makes the gate competitive on candidates. (ii) The calibration that H2 confirms holds on the bank's own actions and fails on actions the bank did not take, which are exactly the ones an agent may propose. (iii) The learned out-of-distribution flag adds nothing over a lookup for unseen action types or unseen argument values, and it escalates a large share of candidates. (iv) Without randomized logging, the model is at chance on candidates and history helps; the useful results above depend on a randomized slice in the log, which a bank would have to introduce deliberately.

FIG. 3 · HISTORY ENCODINGS AGAINST THE MARKOV STATE

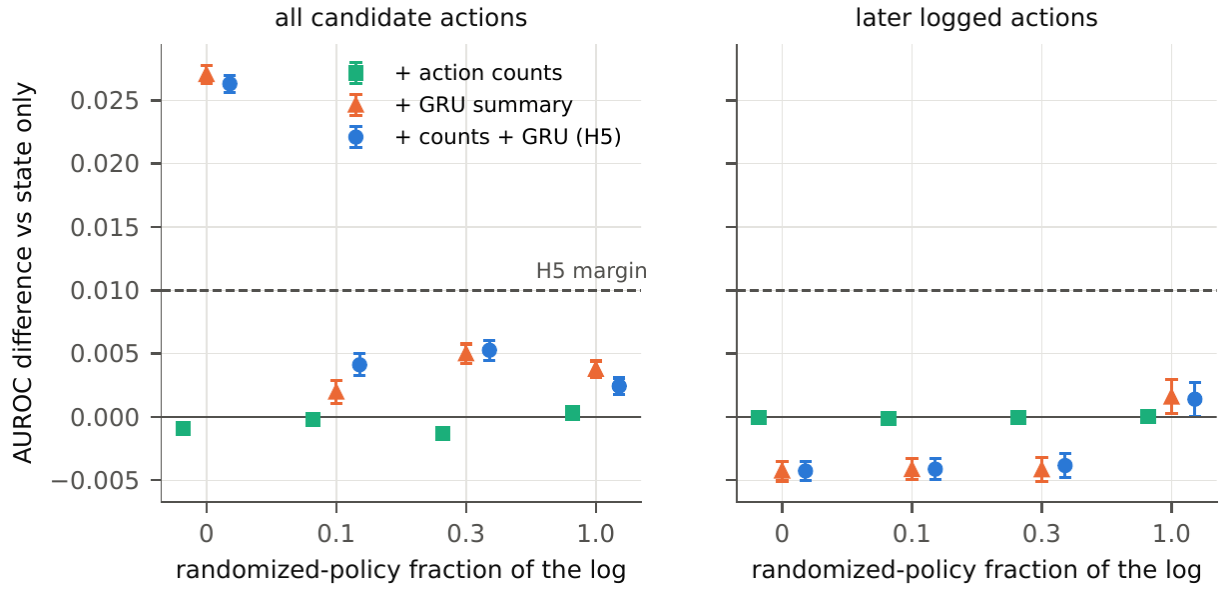


Figure 3. Markov ablation: AUROC with history minus AUROC of the state-only model, with 95 percent cluster-bootstrap intervals (within seeds), by randomized fraction, test slice. The dashed line is H5's equivalence margin (0.01). Left: candidate actions (H5's population). Right: logged actions.

Appendix F Latency and cost

Table 5 gives single-decision latency on one CPU thread (1,000 test decisions after 50 warm-up calls, Apple M5 Max, every native thread pool pinned to one thread; the GRU is timed in its own process). Without the flag our gate is faster than the GRU (0.40 ms at p95); the flag's neighbour query more than doubles its latency. Costs use an assumed price of USD 0.05 per vCPU-hour; this is an order-of-magnitude assumption, not a quote. Our full gate costs more than the GRU, so H3's cost condition also fails; in absolute terms every gate costs a negligible amount per decision. By contrast, DreamGuard's reported 25 ms is GPU latency with a language-model encoder [27] and is not comparable with these CPU numbers.

Table 5. Single-decision latency and cost on one CPU thread (test decisions, seed 0, fraction 0.1). Cost per 1,000 decisions uses an assumed CPU price of USD 0.05 per vCPU-hour, not a quote.

gate	p50 (ms)	p95 (ms)	CPU s per 1,000	USD per 1,000
ours, full gate with OOD flag	0.85	0.91	0.84	1.2×10^{-5}
ours without OOD flag	0.37	0.40	0.37	5.2×10^{-6}
GRU trajectory guardrail	0.46	0.52	0.47	6.5×10^{-6}
DTMC guardrail	1.21	1.74	1.29	1.8×10^{-5}
rules	0.14	0.15	0.14	2.0×10^{-6}

Appendix G Limits specific to E4 and Phase 2b-r

Limits specific to E4. Several properties of the E4 design limit what H6 can show; the numbers below are post hoc (D32) unless stated. (i) *A warm-up that is not steady.* The background review cycle checks every fact once in about 698 to 702 days (measured in the test window), so in the training window many stored facts still carry their onboarding verification and once every fact has been checked none do. In the reviewer's synthetic-seed check described in (iv), the share of facts verified at onboarding at decision time fell from 0.56 in the training window to 0 from day 728 on. The calibration window and the test window therefore differ in distribution, which the conformal guarantee does not cover, and which is consistent with the out-of-distribution flag's high rate on good actions (Section B). (ii) *Pooled thresholds.* One pair of conformal thresholds is calibrated on the good actions of both action kinds together. This controls the pooled rate only and permits different rates by kind; the observed difference by kind is plausibly an effect of pooling, but per-kind calibration was not run, so how much pooling caused is not established. (iii) *Unavoidable rejections.* A share of 0.51 of the failed submissions of the gate without validity scores are rejections that no gate can prevent, which dilutes the registered statistic, and most interventions are spent on them. (iv) *A weak validity signal under calendar-only logging.* The E4 logging design has a calendar review cycle and a randomised slice that reverifies only the facts an action uses; it has no randomised slice of fact checks for the validity model to learn from. The per-decision validity scores are not saved in the E4 outputs, so how informative they were cannot be measured on E4 itself. A check by the adversarial reviewer on synthetic seeds 7 to 9, outside the E4 seeds and not an E4 measurement, found that under this design the validity scores ranked stale facts with an AUROC of 0.60 to 0.62 against 0.67 to 0.69 for fact age alone, and that under uniform random checks at the same budget they reached 0.76 to 0.77 against 0.70 to 0.71 for age. (v) *Escalate equals replan.* The reviewer reverifies the same facts a replan would, at a higher price, and the action then executes; the registered accounting counts escalated decisions as not failed and leaves their outcome costs out, which is not the cost the process incurs (full accounting in Section B). (vi) *An unapproved review cost.* The review cost of 15 and the delay of 3 days were set under the D30 delegation and not reviewed by the human before the test run; every cost comparison involving a gate depends on the review cost. (vii) *Calibration rows after the slice's checks.* Training and calibration rows record the executed state, so for decisions in the randomised slice they hold the freshly reverified facts, a state the gate never sees at decision time; this is a further departure from exchangeability between calibration and deployment. (viii) *The flag's calibration population.* The out-of-distribution threshold is set on all calibration rows, not on good actions, so the flag's rate on good actions is not controlled by its target, and adding its alerts to the conformal alerts gives no guarantee on good actions (Section 4.3). (ix) *Calibration window reused.* The multi-class calibrator is fitted on the same calibration window that then sets the conformal thresholds (and is chosen on validation). The standard split-conformal argument assumes a score fixed before the calibration data are used; multi-class temperature scaling need not preserve the ordering of $1 - P(\text{ok})$ across actions, so this reuse is not covered by it. We measured no coverage defect from it (the conformal part stayed within H2's tolerance), but the argument for the full procedure has not been audited. (x) *Truncated follow-up.* Evaluation stops at the end of the test window, so cases opened near its end are incompletely observed (Section B); all E4 costs are accrued in the window per case opened in it. Per-action-kind (Mondrian) conformal thresholds are future work. Phase 2b-r (Section 6.5) addresses the warm-up part of (i), and (v), (vi), (vii) and (x), for a different pair of rules; it does not re-run the E4 gate.

Limits specific to Phase 2b-r. (i) *Myopic score, transported.* The uplift score estimates the immediate effect of checking an action's used facts, identified from the randomised slice of the logging regime and carried to the deployed arms by assumption; it is not a per-fact value or a full-path value, and deployment differed from the logging regime by a large shift in the stale rate (Section 6.5). (ii) *Thin support.* The randomised treated decisions where a check could prevent a failure, which train m_1, were below the registered minimum for contracts on some pilot and confirmatory seeds, accepted by override. (iii) *The slice is a cost.* Learning the uplift score needs a randomised slice of checks, whose cost per warm-up case was above the smallest worthwhile saving. (iv) *Frozen models, one scenario.* Models were frozen before the test and not refitted; there is one scenario, one primary cell and one transfer condition, and the design was written after the Phase 2 results and two reviews were known. (v) *Pilot decisions.* The burn-in rule was amended after a pilot stop, the slice rate was set by override, and H6r was moved to reported for lack of power; the burn-in and support rules read simulator truth. After the test, with its results read, the failed substitute drift comparison was replaced by the registered logging-regime check (LIV D70); this changes no reported contrast. (vi) *No verdict.* With power that low, the absence of a detected difference between the uplift and risk rules is uninformative; the cost increase against no intervention is a reported contrast, not a tested hypothesis. (vii) *Registered quantities not computed.* The myopia diagnostic (R10), which would set the uplift rule's estimated immediate saving against its realised full-path saving, was not computed. Of the support and transport report (R9), the support counts, the unneeded and missed rates and the stale rate at the decisions of the `fresh_case` arm without a gate are reported; the decision-state comparison for the other arms, P(valid) at decisions and the validity scorer's calibration on the test window were not computed.

FlowX.AI

FlowX.AI builds and orchestrates enterprise-grade AI agents for regulated industries, with a focus on banking, financial services, and insurance. The study behind this paper was run on simulators only; nothing in it was tested on the FlowX.AI Platform, on a FlowX.AI process or on client data.

FlowX.AI Technical Paper · CASS version 1.0.0 · doi.org/10.5281/zenodo.24077491 · Bogdan Răduță, Head of Research · bogdan.raduta@flowx.ai

© 2026 FlowX.AI. Text and code licensed under the Apache License, Version 2.0. Preprint, not peer reviewed. The archived release at the DOI above holds the LaTeX edition of this paper and the complete source.