RISK BEFORE THE ACTION: WHAT PRE-ACTION HIDDEN STATES PREDICT IN LLM AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

An LLM agent may issue a tool call that returns an error. It may also fail its task without any tool error. We study what hidden states predict before the agent generates a response. Our main target is tool rejection, meaning an error reply from the environment. We fit linear probes on frozen hidden states from two Qwen3 model sizes in the retail and airline domains of τ^2-bench. Task-grouped cross-validation shows gains over the tested text baselines in rejection ranking across all four settings. On unseen tasks, frozen 14B probes reach AUROC 0.910 in retail and 0.857 in airline. These scores compare calls within the same task and call position. Repeated sampling from fixed conversations and environment states shows little variation in first-call rejection, yet probe probabilities remain imperfect. We next fit separate 14B probes for rejection and final task failure using the same pre-action layer and prefixes. In cross-validation on training tasks, rejection ranking is stronger on shared task and call-position groups. Retail AUROC is 0.882 for rejection and 0.549 for final failure. We also show when a fixed predictor using only binary calling and rejection records cannot give accurate task-success probabilities across populations. Finally, the tested verification policy shows no detectable advantage over random, observable, or timing-matched triggers. Local risk prediction, task-success estimation, and intervention value therefore require separate evidence.

1 INTRODUCTION

Consider an LLM agent handling a return request. It may send a tool call with an order identifier that does not exist. The tool returns an error, which we call a *rejection*. A risk estimate available before the response could flag this decision for a check. Yet an agent can also receive no tool errors and still miss a required update. Predicting tool rejection and predicting task success therefore address different parts of reliability.

Prior work has found error signals in the internal representations of generated text (Chen et al., 2024; Orgad et al., 2025). Other work estimates feasibility before generation (Chen et al., 2026) or predicts eventual agent failure after an action has been written (Ruan et al., 2026). Agent uncertainty methods also combine signals across steps or use them to guide tool use (Zhao et al., 2025; Han et al., 2024; Suri et al., 2026; Oh et al., 2026). We examine tool rejection before the response is generated, then test what this local signal supports beyond the next call.

We read the hidden state at the last prompt token, before any new response token exists. The chosen tool, its arguments, and the tool reply are all still unknown. We fit a *linear probe* which acts like a lightweight classifier, while keeping the language model fixed. The study uses two sizes of Qwen3 in the retail and airline domains of τ^2-bench. We distinguish the probability of issuing a tool call from the conditional probability of rejection given that a call is issued. This separates tool-use decisions from execution outcomes.

We first measure how well the probe ranks rejected calls above accepted ones using AUROC (Oh et al., 2026). Our score compares calls within the same task and at the same call position. The gain over a text predictor available at the same moment ranges from $+0.097$ to $+0.152$. All four 95% intervals exclude zero. Frozen 14B probes are also evaluated on independent test tasks. Their matched AUROC reaches 0.910 in retail and 0.857 in airline. After the full call is written, we

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

compare hidden states with a predictor using the call text and other visible features. Neither 14B setting shows a detectable gain in this comparison.

We also check whether the predicted probabilities agree with observed event rates. We measure *probability calibration* (Gneiting & Raftery, 2007; Guo et al., 2017; Raghu et al., 2026) in both domains. Repeated sampling from fixed conversations and environment states gives a closer view of these probabilities. The sampled rejection outcomes vary little, yet the probe probabilities remain imperfect.

We next compare rejection and final task failure using separate probes on the same pre-action representation. This additional 14B analysis uses task-grouped cross-validation on training data. On shared task and call-position groups, the tested layer supports stronger rejection ranking. In retail, its advantage over text is also larger for rejection. We then examine binary calling and rejection records. Rejection-free execution relates to task success in opposite directions across domains. The same record can occur in successful and failed runs. We construct populations with identical record distributions but different success rates. A fixed predictor based only on these records gives the same average prediction in both populations.

Finally, we test whether the risk signal helps the agent use a verification instruction. A *gate* triggers the instruction when the risk score exceeds a fixed threshold. The instruction remains active until the task ends. On held-out tasks, this policy shows no detectable advantage over random triggers or a gate based on conversation history. We also match random trigger times to those of the hidden gate, reducing differences in how long the instruction remains active. This control also shows no detectable advantage for the hidden gate. This finding concerns the tested instruction and threshold. It shows why the value of an intervention needs evidence beyond the quality of its risk predictions.

Our contributions center on three findings:

(1) Pre-action probes outperform the tested text baselines in rejection ranking. The gains hold across all four settings, and frozen 14B probes also rank rejection well on unseen tasks.

(2) At the same pre-action layer, the tested probes rank rejection better than final task failure. Separate theory shows when a fixed predictor using only binary calling and rejection records cannot give accurate task-success probabilities across populations.

(3) The tested risk-triggered verification policy shows no detectable advantage over its controls. These include random triggers, a trigger based on visible conversation history, and a random trigger matched on timing.

2 Related work

Uncertainty in interactive agents. Agent uncertainty depends on interaction and prediction scope (Oh et al., 2026; Kirchhof et al., 2025). SAUP combines step-level uncertainty (Zhao et al., 2025), while RUPA also models dependencies and task goals (Ma et al., 2026). Our bounds concern binary calling and rejection records, which omit such richer information. UALA guides tool use (Han et al., 2024), and SAGE selects clarification questions (Suri et al., 2026). We test rejection-triggered verification against random, observable, and timing-matched controls.

Internal signals and prediction timing. Prior work probes hidden states for hallucinations (Chen et al., 2024; Orgad et al., 2025) and instruction-following errors (Heo et al., 2025a;b). Other approaches use layer-update geometry (Eusebi et al., 2026) or estimate answerability before generation (Chen et al., 2026). Doomed predicts eventual failure from the final token of generated actions and supports early abort (Ruan et al., 2026). Last Step Matters finds weak early failure prediction from verbal confidence, token probabilities, and consistency in deep-research agents (Li et al., 2026). We predict tool rejection before generation and compare hidden states with baselines available at the same time.

Sampling and probability evaluation. Semantic entropy measures variation across sampled meanings (Farquhar et al., 2024), and Semantic Entropy Probes estimate it from hidden states (Kossen et al., 2024). Consistency and correctness remain distinct (Xiao et al., 2025; Mehta, 2026). We repeat sampling at fixed contexts and environment states to study rejection probabilities. Log

and squared-error scores assess probability quality beyond ranking (Gneiting & Raftery, 2007). Trajectory Proper Score evaluates eventual-success probabilities along a trajectory (Raghu et al., 2026); our probability analysis targets tool rejection.

3 Problem Setup and Evaluation

3.1 Prediction targets

An assistant turn is one response generated by the agent. A trajectory is the sequence of turns and environment replies until the task ends. At turn t, let C_t contain the conversation, policy text, and tool definitions visible to the agent. Let S_t be the environment state, which is only partly visible. The agent draws a response A_t from a fixed policy.

We set $J_t = 1$ when the interface records at least one tool call in the response. We set $Z_t = 1$ if any resulting tool reply has an error flag. Otherwise, $Z_t = 0$, including turns without a call. Each turn contributes one observation. Thus, Z_t measures tool rejection. An accepted call may still be irrelevant to the task or leave a requirement unmet. Call text that the interface does not execute falls outside this rejection target.

We study three probabilities under this policy and the evaluated distribution of tasks and environment states:

$$r_1(c) = \mathbb{P}(J_t = 1 \mid C_t = c), \qquad \text{a call occurs,} \tag{1}$$

$$r_2(c) = \mathbb{P}(Z_t = 1 \mid J_t = 1,\, C_t = c), \qquad \text{rejection given a call,} \tag{2}$$

$$r_3(c) = \mathbb{P}(J_t = 1, Z_t = 1 \mid C_t = c), \qquad \text{a call occurs and is rejected.} \tag{3}$$

The conditional probability rule gives $r_3(c) = r_1(c)r_2(c)$ whenever $r_1(c) > 0$. We fit separate probes and do not force their estimates to satisfy this identity. Our main ranking comparison targets r_2 and restricts evaluation to turns that issue a call. Online gating uses r_3, which is defined for every assistant turn before the call decision is known.

Let $Y \in \{0, 1\}$ denote task success according to the benchmark evaluator. It depends on the full interaction and the task requirements. Local rejection Z_t and terminal success Y are different labels.

3.2 Observation times and predictors

Let $H_t^{(\ell)}$ be the hidden state at layer ℓ and the chosen observation time. Our main readout is at p_0, the final prompt token, before any new response token. We call this the *pre-action* readout. The upcoming tool, its arguments, and its reply are still unknown. Later readouts follow the tool name (p_2), the completed arguments (p_5), and the call message (p_6). They include generated action content before the tool reply. Figure 1 summarizes the main definitions. Appendix B.2 defines all positions from p_0 to p_6.

The probe is an L2-regularized logistic classifier on frozen hidden states. At p_0, we compare it with predictors using information already available in the conversation. $M1h$ uses observable history features, such as earlier calls and errors. $T1$ uses word features of the preceding conversation. $T2$, our primary text baseline, uses character features from its last 800 characters. The auxiliary stack combines their scores. The text baselines omit the policy text and tool definitions, which are constant within each domain. At p_6, the comparison instead uses an observable predictor that also reads the call. All detailed settings are provided in Appendix B.3.

For a fixed model, the pre-action hidden state is a function of the visible context. A gain over a text baseline concerns the tested predictors and feature sets. It does not establish that the full context lacks the relevant information or that the model holds an explicit internal belief.

3.3 Data and model setup

The main experiments use Qwen3-14B-AWQ with thinking mode off and temperature 0.7 (Yang et al., 2025). The benchmark user simulator uses the same model. We study the retail and airline domains of τ^2-bench separately (Barres et al., 2025). Their training archives contain $1{,}352$ and 539 trajectories from 74 and 30 tasks, respectively. These archives contain $12{,}828$ retail and $3{,}802$ airline

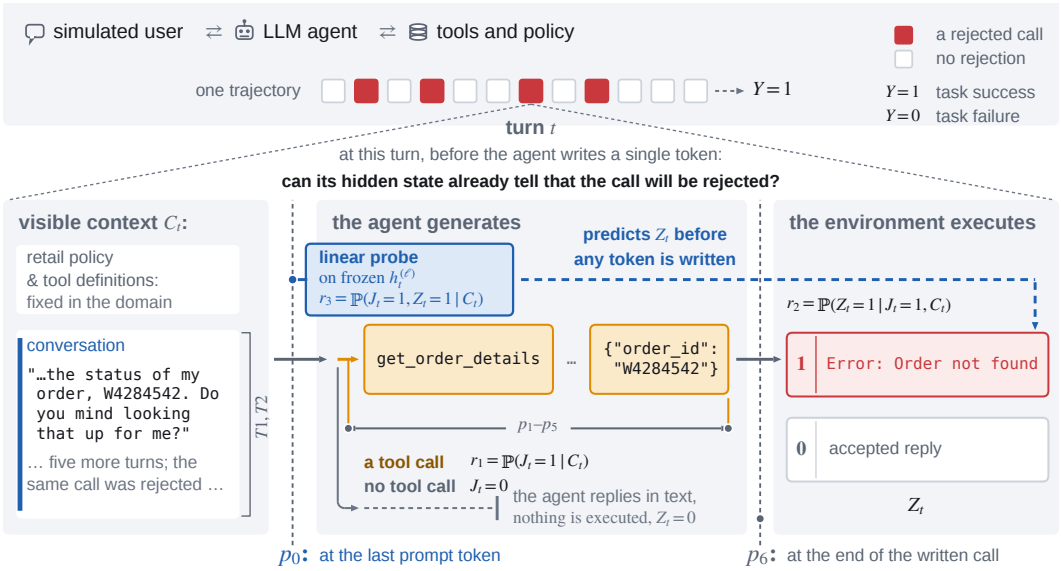

Figure 1: Prediction targets and observation times along one assistant turn. The example trajectory succeeds even though some of its calls are rejected.

turns with a tool call. The official test sets contain 40 retail and 20 airline tasks. Eight seeds per task give 320 and 160 test trajectories. We also test Qwen3-8B-AWQ, with its setup and the shared probe settings reported in Appendix B.4.

3.4 Evaluation protocol

Task splits and calibration. Primary training-stage comparisons use five folds grouped by task. All trajectories from a task stay in the same fold. Each out-of-fold (OOF) prediction comes from a model fitted on the other folds. Feature scaling and text vocabularies are fitted within the training folds. For independent test evaluation, probes are fitted on all training tasks and applied without further changes. A logistic calibration map adjusts the probabilities using training OOF scores. Neither probe fitting nor calibration uses test tasks. Appendix B.5 gives the calibration folds and the order of fitting and evaluation.

Ranking within task and call position. For each target, pooled AUROC compares positive and negative observations across its full evaluation set. It may therefore reflect differences in task difficulty or trajectory progress. Our main r_2 comparison instead matches task and exact call position. Call position counts assistant turns that issue a call; text-only turns do not advance it. For each matched group g, let n_g^+ and n_g^- count rejected and accepted observations. We retain only groups containing both labels and report

$$\mathrm{AUC}_{\mathrm{task,slot}} = \frac{\sum_g n_g^+ n_g^- \, \mathrm{AUC}_g(s)}{\sum_g n_g^+ n_g^-}, \tag{4}$$

where s is the prediction score and ties count as one half. This gives equal weight to each eligible pair of a rejected and an accepted call. Matching task and call position does not fix the full conversation. Groups containing only one label are excluded from this ranking measure. Appendix B reports alternative weightings and position groupings, interval estimation, permutation checks, and analysis status.

4 Pre-action risk prediction

We first test whether hidden states predict tool rejection before response generation. We then examine whether the probe scores give reliable probabilities.

4.1 PREDICTING TOOL REJECTION

Comparison with pre-action baselines. Table 1 compares the hidden probe with predictors that use information available at the same time. All results use OOF predictions on training tasks and compare rejected and accepted calls within the same task and exact call position. The 14B probe reaches matched AUROC 0.881 in retail and 0.779 in airline. Its paired gains over the primary text baseline $T2$ are $+0.127$ and $+0.097$. Separate 8B probes, fitted with the same settings, show gains of $+0.126$ and $+0.152$. All four 95% intervals exclude zero. The training-stage permutation checks give $p = 0.001$ in all four settings. This supports pre-action rejection prediction across two sizes of one model family and two domains.

Table 1: Tool-rejection ranking, hidden state gain and 95% interval before response generation, using OOF predictions on training tasks.

Predictor	**Retail**		**Airline**	
	14B	8B	14B	8B
History features $M1h$	0.708	0.721	0.647	0.601
Word features $T1$	0.738	0.711	0.643	0.638
Character features $T2$	0.755	0.750	0.682	0.631
Auxiliary stack	0.767	0.774	0.689	0.630
Hidden state p_0	**0.881**	**0.876**	**0.779**	**0.783**
$p_0 - T2$ (primary)	$+0.127$	$+0.126$	$+0.097$	$+0.152$
95% interval	$[0.105, 0.149]$	$[0.098, 0.155]$	$[0.050, 0.142]$	$[0.076, 0.218]$
$p_0 -$ stack (auxiliary)	$+0.115$	$+0.103$	$+0.091$	$+0.153$

Frozen probes on unseen tasks. We fit each 14B probe on all training tasks and freeze it. We then evaluate these probes on independent test tasks. Matched AUROC is 0.910 with interval $[0.887, 0.933]$ in retail and 0.857 with interval $[0.771, 0.909]$ in airline. Alternative group weightings and coarser call positions give similar results (Appendix B.7). Both permutation checks give $p = 0.001$. This tests generalization of the hidden probe; the paired gains over $T2$ above come from training-task OOF comparisons.

Comparison after the call is written. We compare predictors at p_6 in Figure 1, after the full call is written and before execution. The observable predictor uses the call text and other features available at this point. Training-task OOF comparisons show no detectable hidden-state gain for the 14B model in either domain. Both paired 95% intervals include zero. The baselines differ across times, and we have not directly tested the difference between the gains. Appendix B.8 reports the scores and intervals.

4.2 PROBABILITY QUALITY UNDER REPEATED SAMPLING

We next test whether the scores that rank tool rejection also give accurate probabilities. We first evaluate calibration on unseen tasks, then examine repeated outcomes at fixed contexts and states.

Probability diagnostics on unseen tasks. The calibration maps are fitted on training OOF scores and frozen before testing. Their positive slopes preserve AUROC. Table 2 reports calibration slope and expected calibration error (ECE). A well-calibrated score has slope one. ECE measures the average absolute gap between predicted and observed rates in score bins (Van Calster et al., 2019). The AUROC pools all valid turns for each target. Calibration moves all six slopes toward one. ECE falls for all retail targets and airline r_1, but rises for airline r_2 and r_3. Thus, improvement depends on the diagnostic. Appendix B.10 gives their definitions and scope.

Repeated outcomes at a fixed context and state. Aggregate calibration does not show how outcomes vary at one fixed context and state. We next audit OOF scores in retail and airline calling-turn panels from the 14B training tasks. Each panel draws 16 responses at each of 500 prefixes. Each draw restores (C, S) and executes the first parsed call. This audits first-call rejection; the original

Table 2: Ranking and probability diagnostics on held-out 14B trajectories.

Domain	Target	Pooled AUROC	Slope raw → cal.	ECE raw → cal.
Retail	r_1	0.986	0.778 → 1.034	0.0125 → 0.0055
	r_2	0.934	0.661 → 1.056	0.0371 → 0.0125
	r_3	0.948	0.729 → 1.042	0.0191 → 0.0115
Airline	r_1	0.969	0.612 → 0.872	0.0384 → 0.0315
	r_2	0.912	0.678 → 1.129	0.0353 → 0.0375
	r_3	0.923	0.687 → 1.103	0.0255 → 0.0257

turn-level label covers any rejected call. Weighted estimates show no observed rejection variation on 96.9% and 96.0% of eligible prefixes, respectively. Finite sampling still allows rare errors. Appendix B.11 reports the additional retail all-turn panel, eligible sets, and weights.

Probability scores under repeated sampling. Stable sampled outcomes do not ensure accurate probe probabilities. We evaluate log loss on sampled outcomes and squared error against their rejection rates (Gneiting & Raftery, 2007; Raghu et al., 2026). For the latter, a variance correction adjusts for finite sampling where at least two calls are available. Calibration lowers both point estimates in the calling-turn panels. Corrected squared error falls from 0.0855 to 0.0826 in retail and from 0.0921 to 0.0854 in airline. Appendix B.11 reports all scores and additional calibration diagnostics, with a visual summary in Figure 3.

Interpreting the remaining error. Proposition 1 connects error against fixed-state probabilities to Brier loss, the expected squared error for a binary outcome.

Proposition 1 (Brier error with partially observed state)**.** *Fix a binary outcome $B \in \{0,1\}$ and an evaluation distribution. Let $q(C,S) = \mathbb{E}[B \mid C,S]$ and $r(C) = \mathbb{E}[B \mid C]$. For any fixed predictor $f(C) \in [0,1]$,*

$$\mathbb{E}[(B - f(C))^2] = \underbrace{\mathbb{E}[q(1-q)]}_{\substack{\text{outcome variation} \\ \text{at fixed } (C,S)}} + \underbrace{\mathbb{E}[(q-r)^2]}_{\substack{\text{state variation} \\ \text{given } C}} + \underbrace{\mathbb{E}[(r - f(C))^2]}_{\substack{\text{probe error} \\ \text{relative to } r(C)}}. \tag{5}$$

For conditional-call rejection, use the distribution restricted to calls.

Repeated draws at restored (c,s) estimate $q(c,s)$; $r(c)$ instead averages over states given the visible context. Under a common outcome and population, squared error against q combines the last two terms. Our panels cannot separate them or show that the state term is positive. Low observed sampling variation therefore does not establish that better probes can remove all remaining error. Appendix A.1 gives the proof, target-specific conditioning, and the result for observing the generated call.

5 From Local Execution to Task Success

Section 4 establishes a signal for the next tool error. We first compare this target with final task failure using the same pre-action representation. We then examine what local execution records support. The analyses below use the 14B training benchmark, with 1,352 retail trajectories and 539 airline trajectories.

5.1 From tool rejection to task outcomes

Two targets at the same prefix. We fit separate linear probes for rejection Z_t and final failure $1 - Y$ on the same calling-turn prefixes. Both read layer 25 at p_0, chosen in the earlier rejection analysis. They share task-grouped folds, trajectory weights, and tuning budgets. This additional analysis uses only training trajectories. We evaluate task and exact-call-position groups containing both labels for both targets. We average group AUROCs within each task, then average tasks equally. This gives both targets the same weights, unlike the pair weighting in Eq. (4). Retail is the primary analysis; airline is descriptive because only 12 tasks contribute.

Table 3: AUROC and paired contrasts for two targets at p_0. Brackets give 95% task-bootstrap intervals.

Measure	Retail	Airline (descriptive)
Hidden: rejection	0.882	0.818
Hidden: final failure	0.549	0.586
Rejection minus failure	$+0.334$ $[0.294, 0.374]$	$+0.232$ $[0.106, 0.365]$
Hidden minus text: rejection	$+0.179$ $[0.146, 0.214]$	$+0.159$ $[0.013, 0.322]$
Hidden minus text: final failure	$+0.025$ $[-0.021, 0.073]$	$+0.022$ $[-0.171, 0.171]$
Difference in gains	$+0.154$ $[0.097, 0.212]$	$+0.137$ $[-0.057, 0.341]$

Shared samples: retail 6,092 prefixes from 999 trajectories and 55 tasks; airline 501 prefixes from 215 trajectories and 12 tasks. They cover 47.5% and 13.2% of calling prefixes, respectively.

A larger advantage for local prediction. Rejection ranking is stronger on the shared groups (Table 3). Retail also shows a larger hidden advantage over text for rejection, supported by the direct gain contrast. Neither domain shows a detectable gain over text for final failure. This does not establish equivalence or absence of task information. The result concerns one layer and linear probes; all hidden fits selected the strongest tested regularization. Appendix B.12 reports fitting details and scope, with a visual comparison in Figure 4.

No rejections do not imply task success. Let $K = \sum_{t=0}^{T} Z_t$ count turns with a tool rejection, where T is the last assistant turn. We call a trajectory *clean* when $K = 0$. This only means that no tool call returned an error, but the agent may still have missed a required action. We compare success rates between clean trajectories and those with at least one rejection.

Table 4 separates three ways to make this comparison. Within tasks, the difference is positive in retail and negative in airline, with both intervals excluding zero. This average gives equal weight to tasks containing both kinds of trajectory. The pooled comparison instead gives equal weight to trajectories, and its airline interval includes zero. These are different estimates of association, not interchangeable tests of the same quantity.

Table 4: Task success rate in clean trajectories minus that in trajectories with a rejection. Brackets give 95% task-cluster bootstrap intervals.

Comparison	Retail	Airline
Pooled	$+0.211$ $[0.064, 0.348]$	-0.172 $[-0.370, 0.031]$
Within task	$+0.103$ $[0.011, 0.201]$	-0.138 $[-0.253, -0.027]$
Within task and write count	$+0.088$ $[-0.008, 0.186]$	-0.011 $[-0.125, 0.139]$

Activity offers one possible explanation. Clean trajectories have fewer write-call turns: 1.05 versus 2.51 in retail, and 2.53 versus 3.54 in airline. A write call attempts to change the environment, such as updating a booking. Fewer attempts leave fewer opportunities for rejection, but may also leave required work undone. Comparing trajectories within the same task and write count moves both point estimates toward zero. Write count is measured after generation, and this comparison also changes the eligible groups. It therefore does not identify a causal mechanism. Appendix B.13 gives the group counts and measurement details; Figure 5 illustrates the associations and a shared record.

An evaluator-based action oracle and a later probe for reference-write completion also show task-related information in richer inputs (Appendix B.14).

5.2 What local records can support

We now ask when local outcomes alone support a shared success predictor. We consider only K or the binary execution record $L = [(J_0, Z_0), \ldots, (J_T, Z_T)]$. The record keeps turn order but omits

action content and task requirements. Hidden states and predicted risk sequences may retain more context, so the bounds do not cover them in general.

A shared probability from rejection counts. For $K = 0$, observed success is 0.447 in retail and 0.141 in airline. The difference is 0.306, with a 95% interval of $[0.106, 0.486]$ from a post-hoc task bootstrap (Appendix B.13). This compares the same count across domains, complementing the within-domain contrasts in Table 4.

For domain $D = d$, let $\alpha_d(k) = \mathbb{P}(Y = 1 \mid K = k, D = d)$. A shared predictor $g(K)$ matches each supported count group's success probability when

$$\mathbb{E}[Y - g(K) \mid K = k, D = d] = 0. \tag{6}$$

Proposition 2 (Shared probabilities from rejection counts)**.** *A shared g satisfying Eq. (6) in two domains exists if and only if their $\alpha_d(k)$ agree at every count supported by both. At any shared count k, every g has absolute conditional mean error at least $|\alpha_d(k) - \alpha_{d'}(k)|/2$ in one domain.*

The zero-rejection estimates provide evidence against such a shared probability. This requirement is stronger than ordinary score calibration, which pools counts assigned the same score.

Keeping turn order does not ensure transfer. We next retain turn order in L, while keeping the same predictor across populations.

Proposition 3 (Transfer from a fixed execution record)**.** *Let $\mathbb{P}$ and $\mathbb{P}'$ be two trajectory distributions with the same distribution of L. Set $\Delta = |\mathbb{E}_{\mathbb{P}} Y - \mathbb{E}_{\mathbb{P}'} Y|$. Every fixed measurable predictor $\phi(L) \in [0, 1]$ satisfies*

$$\max_{\mathbb{Q} \in \{\mathbb{P}, \mathbb{P}'\}} |\mathbb{E}_{\mathbb{Q}} \phi(L) - \mathbb{E}_{\mathbb{Q}} Y| \geq \frac{\Delta}{2}. \tag{7}$$

Identical input distributions give the predictor the same mean output in both populations. Its excess Brier loss is also at least $\Delta^2/4$ in one population, relative to the best predictor using L there.

For example, $L = [(0, 0), (1, 0), (0, 0)]$ occurs in 17 successful and 17 failed retail runs. Within each domain, we select successful and failed runs from disjoint task groups. Reweighting their shared records gives identical L distributions and $\Delta = 1$. The construction uses outcomes for selection and need not preserve each task's rollout distribution. The original retail and airline corpora do not satisfy the equal-distribution premise. Appendix A.2 gives proofs and construction details. Useful prediction within a fixed population remains possible.

6 Risk-triggered Verification

We test whether risk-based triggering adds value to a fixed verification instruction.

6.1 Intervention and controls

A fixed instruction after the first trigger. The hidden gate reads the frozen all-turn score r_3 before generation. At the first threshold crossing, we add an instruction to verify relevant entities and their state through a read-only lookup. It remains active until the trajectory ends. Scores are computed on held-out baseline prefixes. At a trigger, we restore the history and environment state, then sample a new continuation. The 14B study covers 320 retail trajectories from 40 tasks and 160 airline trajectories from 20 tasks. Arms share baseline trajectories, with separate continuation seeds. Untriggered trajectories keep their original outcomes.

What each control tests. The original agent is $A1$, and hidden-triggered verification is $A2$. Random triggering ($A3$) matches $A2$'s trajectory-level firing rate in expectation and selects a baseline turn uniformly. The observable gate ($A4$) scores conversation text and history features. The $A2$ and $A4$ thresholds are fixed at a 10% training alert rate. Their test trajectory-level firing rates can differ. $A6$ branches at the same prefixes as $A2$ without verification, comparing the instruction with resampling alone. Appendix B.15 gives the full protocol and first-turn verification control ($A5$).

Table 5: $A2$ minus each control on 320 held-out retail trajectories, with 95% task-bootstrap intervals. Negative rejection differences and positive success differences favor $A2$; pp denotes percentage points.

Control	**Rejections / trajectory**	**Task success** (pp)
Random trigger ($A3$)	-0.163 $[-0.441, 0.100]$	-0.9 $[-5.6, 4.1]$
Observable trigger ($A4$)	$+0.072$ $[-0.172, 0.303]$	$+0.3$ $[-5.3, 6.3]$
Resampling only ($A6$)	-0.009 $[-0.206, 0.191]$	$+1.6$ $[-2.8, 6.3]$
Timing-matched random ($A7$)	-0.134 $[-0.391, 0.110]$	$+4.4$ $[-0.6, 9.4]$

6.2 Results and interpretation

Direct comparisons show no detectable advantage. The primary endpoint counts turns with a tool rejection per retail trajectory. Task success is secondary, with pooled and descriptive domain-level estimates. Airline rejection results are descriptive. We average paired differences over all baseline trajectories and bootstrap tasks. Every interval in Table 5 includes zero. They allow both benefit and harm and do not establish equivalence.

Retail success improves by 4.7 percentage points for $A2$ over $A1$, while random triggering ($A3$) gains 5.6 points. Improvement over the original agent therefore does not isolate the value of risk-based selection. Appendix B.15 reports all arm-level and airline results, with direct comparisons in Figure 6.

Reducing the difference in trigger timing. The hidden gate typically fires earlier than $A3$. Since verification persists, this changes how long the instruction can act. We specified the additional random control $A7$ after the original analysis but before generating its continuations. It uses $A3$'s attempted-trigger trajectories and samples relative positions from $A2$'s distribution within each domain. Mean remaining baseline turns in retail are 16.5 for $A2$, 8.2 for $A3$, and 15.0 for $A7$. This narrows the timing gap without matching exposure in every trajectory. $A2$ still shows no detectable advantage over $A7$ on either endpoint. Appendix B.16 gives matching details and the exploratory timing comparison; Figure 7 shows the trigger distributions.

Prediction quality and intervention value. A post-hoc comparison evaluates r_3 on all held-out retail baseline turns. At a matched alert rate, precision is 0.857 for hidden scores and 0.628 for observable scores. This does not establish prediction quality at first triggers or after intervention. Its observable threshold also differs from $A4$'s frozen threshold (Appendix B.17).

Let F denote triggering under a fixed gate. Let U^g, U^0 be full-trajectory utilities under the gated and original policies. Assume untriggered outcomes agree and $\mathbb{P}(F) > 0$. With cost κ per trigger, the net gain is

$$\Delta V = \mathbb{E}[U^g - U^0 - \kappa \mathbb{1}_F] = \mathbb{P}(F)\big(\mathbb{E}[U^g - U^0 \mid F] - \kappa\big). \tag{8}$$

Risk ranking does not determine whether verification improves the selected continuations. The identity clarifies this distinction but does not identify the cause of the observed result (Appendix A.3).

These results concern one instruction and one training alert level in two domains. Appendices B.15 and B.18 report failed-branch sensitivity and the post-hoc completion-token cost analysis (Figure 8).

7 Conclusion

Pre-action hidden states support tool-rejection prediction in the agents we study. Task-grouped comparisons show gains over the tested text baselines, and frozen probes retain strong ranking on unseen tasks. An additional comparison on training tasks finds stronger rejection ranking than final-failure ranking from the same pre-action layer. Our theory specifies when local execution records cannot support shared task-success probabilities. The tested verification policy shows no detectable advantage over its controls. These results call for separate evidence for local risk prediction, probability quality, task-success estimation, and intervention value. Further discussion and limitations appear in Appendix E.

AI USE STATEMENT

In this work, we used generative AI tools for improving the language and presentation of this manuscript, including grammar correction, proofreading, and wording refinement. No AI tools were used to generate or develop the research ideas, methodology, experiments, analysis, results, or conclusions. The authors remain fully responsible for all content of the manuscript. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we provide detailed descriptions of the proposed method, training procedures, datasets, evaluation protocols, and implementation details in the main paper and supplementary material. All datasets used in our experiments are publicly available from their original sources. We will release the source code, configuration files, and scripts required to reproduce the reported experiments upon publication.

REFERENCES

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016. doi: 10.48550/arXiv.1610.01644.

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2-bench: Evaluating conversational agents in a dual-control environment. *arXiv preprint arXiv:2506.07982*, 2025.

Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. INSIDE: LLMs' internal states retain the power of hallucination detection. In *International Conference on Learning Representations*, 2024.

Lihu Chen, Gerard de Melo, Fabian M. Suchanek, and Gaël Varoquaux. Query-level uncertainty in large language models. In *International Conference on Learning Representations*, 2026.

Aliai Eusebi, Alexander Herzog, Xiaoyu Liang, Marie Vasek, Enrico Mariconti, and Lorenzo Cavallaro. Reading calibrated uncertainty from language model trajectories. *arXiv preprint arXiv:2605.22864*, 2026.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630:625–630, 2024. doi: 10.1038/s41586-024-07421-0.

Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007. doi: 10.1198/016214506000001437.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1321–1330. PMLR, 2017.

Jiuzhou Han, Wray Buntine, and Ehsan Shareghi. Towards uncertainty-aware language agent. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 6662–6685. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.398.

Juyeon Heo, Christina Heinze-Deml, Oussama Elachqar, Kwan Ho Ryan Chan, Shirley Ren, Andrew Miller, Udhyakumar Nallasamy, and Jaya Narain. Do LLMs "know" internally when they follow instructions? In *International Conference on Learning Representations*, 2025a.

Juyeon Heo, Miao Xiong, Christina Heinze-Deml, and Jaya Narain. Do LLMs estimate uncertainty well in instruction-following? In *International Conference on Learning Representations*, 2025b.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*,

pp. 2733–2743, Hong Kong, China, 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1275.

Michael Kirchhof, Gjergji Kasneci, and Enkelejda Kasneci. Position: Uncertainty quantification needs reassessment for large language model agents. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 81665–81677. PMLR, 2025.

Jannik Kossen, Jiatong Han, Muhammed Razzak, Lisa Schut, Shreshth Malik, and Yarin Gal. Semantic entropy probes: Robust and cheap hallucination detection in LLMs. *arXiv preprint arXiv:2406.15927*, 2024.

Zongyue Li, Chengyue Yu, Lei Zang, Chenyi Zhuang, Linjian Mo, and Leilei Gan. Last step matters: Early uncertainty cannot predict failure in long-horizon agents. *arXiv preprint arXiv:2608.29685*, 2026.

Zhengzhao Ma, Boxi Cao, Yaojie Lu, Hongyu Lin, Xianpei Han, and Le Sun. From sequence to structure: Relational uncertainty propagation for LLM agents. *arXiv preprint arXiv:2608.16002*, 2026.

Aman Mehta. When agents commit too soon: Diagnosing premature commitment in LLM agents. *arXiv preprint arXiv:2606.22936*, 2026.

Changdae Oh, Seongheon Park, To Eun Kim, Jiatong Li, Wendi Li, Samuel Yeh, Xuefeng Du, Hamed Hassani, Paul Bogdan, Dawn Song, and Sharon Li. Uncertainty quantification in LLM agents: Foundations, emerging challenges, and opportunities. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16219–16250. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.738.

Hadas Orgad, Michael Toker, Zorik Gekhman, Roi Reichart, Idan Szpektor, Hadas Kotek, and Yonatan Belinkov. LLMs know more than they show: On the intrinsic representation of LLM hallucinations. In *International Conference on Learning Representations*, 2025.

Suresh Raghu, Satwik Pandey, and Shashwat Pandey. Proper scoring rules for agentic uncertainty quantification. *arXiv preprint arXiv:2605.24756*, 2026.

Kai Ruan, Zihe Huang, Ziqi Zhou, Qianshan Wei, Jinghao Lin, Xuan Wang, and Hao Sun. Doomed from the start: Early abort of LLM agent episodes via a recall-controlled probe cascade. *arXiv preprint arXiv:2607.06503*, 2026.

Manan Suri, Puneet Mathur, Nedim Lipka, Franck Dernoncourt, Ryan A. Rossi, and Dinesh Manocha. Structured uncertainty guided clarification for LLM agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 40811–40838. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.2028.

Ben Van Calster, David J. McLernon, Maarten van Smeden, Laure Wynants, Ewout W. Steyerberg, and Topic Group Evaluating Diagnostic Tests and Prediction Models of the STRATOS Initiative. Calibration: the achilles heel of predictive analytics. *BMC Medicine*, 17(1):230, 2019. doi: 10.1186/s12916-019-1466-7.

Quan Xiao, Debarun Bhattacharjya, Balaji Ganesan, Radu Marinescu, Katya Mirylenka, Nhan H. Pham, Michael Glass, and Junkyu Lee. The consistency hypothesis in uncertainty quantification for large language models. In *Proceedings of the Forty-first Conference on Uncertainty in Artificial Intelligence*, volume 286 of *Proceedings of Machine Learning Research*, pp. 4636–4651. PMLR, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger

Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Qiwei Zhao, Dong Li, Yanchi Liu, Wei Cheng, Yiyou Sun, Mika Oishi, Takao Osaki, Katsushi Matsuda, Huaxiu Yao, Chen Zhao, Haifeng Chen, and Xujiang Zhao. Uncertainty propagation on LLM agent. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6064–6073. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.302.

A PROOFS

A.1 PROBABILITY, STATE, AND OBSERVATION TIME

Proposition 1. All expectations below refer to the same fixed distribution of (B, C, S). Conditioning on (C, S) gives

$$\mathbb{E}[(B - f(C))^2 \mid C, S] = q(1-q) + (q - f(C))^2.$$

The cross term vanishes because $\mathbb{E}[B - q \mid C, S] = 0$. The tower property also gives $\mathbb{E}[q \mid C] = r$. Expanding $q - f = (q - r) + (r - f)$, the remaining cross term satisfies

$$\mathbb{E}[(q - r)(r - f(C))] = \mathbb{E}[(r - f(C))\mathbb{E}[q - r \mid C]] = 0.$$

Taking expectations in the first display proves Eq. (5). The same calculation yields

$$\mathbb{E}[(q - f(C))^2] = \mathbb{E}[(q - r)^2] + \mathbb{E}[(r - f(C))^2]. \tag{9}$$

No assumption that S is fully hidden, or that the state term is positive, is needed. In our restored-state protocol, the environment response is fixed by the state and executed call. The first term then reflects response sampling.

This is an application of conditional expectation and the standard squared-loss decomposition, not a new general impossibility theorem. The minimum context-based Brier loss is $\mathbb{E}[q(1-q)] + \mathbb{E}[(q - r)^2]$, attained by $f = r$. A panel at fixed (c, s) can estimate $q(c, s)$ but does not by itself identify the two terms on the right of Eq. (9) separately. Sampling frames, outcome definitions, and weights must match before applying either identity to reported averages. For the conditional-call outcome, apply the proof under the distribution restricted to calls. With the original trajectory labels, $B = J$ gives r_1; $B = Z$ under $\mathbb{P}(\cdot \mid J = 1)$ gives r_2; and $B = Z$ on all turns gives r_3. These mappings do not identify a panel's marginal probability with the corresponding probability in the original trajectory population.

Corollary 1 (Additional action information)**.** *For the same binary outcome B and evaluation distribution, let $r_0(C) = \mathbb{E}[B \mid C]$ and $r_A(C, A) = \mathbb{E}[B \mid C, A]$. Let $\mathcal{B}_0^*$ and $\mathcal{B}_A^*$ be the minimum Brier losses over all predictors using C and (C, A), respectively. Then*

$$\mathcal{B}_0^* - \mathcal{B}_A^* = \mathbb{E}[(r_A - r_0)^2] \geq 0. \tag{10}$$

Proof. The minimizers are r_0 and r_A. Write $B - r_0 = (B - r_A) + (r_A - r_0)$. Since $\mathbb{E}[B - r_A \mid C, A] = 0$, the two terms are orthogonal in squared expectation. Their squared expectations therefore add, which gives Eq. (10). □

The result allows state uncertainty to remain after observing the action. It compares the best predictors over nested information sets. It does not compare particular hidden-state and text predictors, use different outcomes at the two times, or imply monotonicity of their AUROCs.

A.2 LOCAL EXECUTION RECORDS

Proposition 2. Since $g(K)$ is constant on the group $(K = k, D = d)$,

$$\mathbb{E}[Y - g(K) \mid K = k, D = d] = \alpha_d(k) - g(k).$$

Thus Eq. (6) holds exactly when $g(k) = \alpha_d(k)$. On counts shared by the two domains, this is possible precisely when their conditional success probabilities agree. On counts unique to one domain, choose that domain's probability; elsewhere choose any value in $[0, 1]$. For a shared count, the triangle inequality gives

$$|\alpha_d(k) - g(k)| + |\alpha_{d'}(k) - g(k)| \geq |\alpha_d(k) - \alpha_{d'}(k)|.$$

At least one error is at least half the probability gap.

This condition does not require monotonicity in K. It is stronger than ordinary score calibration, $\mathbb{E}[Y \mid g(K), D] = g(K)$. Score calibration pools count groups assigned the same score. For a counterexample, let $K = 0$ and $K = 1$ each have probability $1/2$ in both domains. Let the conditional success probabilities be $(0.8, 0.2)$ in one domain and $(0.2, 0.8)$ in the other. The constant predictor $g(K) = 0.5$ satisfies $\mathbb{E}[Y \mid g(K), D] = g(K)$ in both. It is also non-increasing in K. Yet it fails Eq. (6) in every count group. Opposite associations alone therefore do not rule out ordinary score calibration.

Mean error in Proposition 3. Write μ for the common distribution of L. A fixed function $\phi(L)$ has the same mean $m = \int \phi(\ell)\, d\mu(\ell)$ in both populations. Let $y = \mathbb{E}_{\mathbb{P}} Y$ and $y' = \mathbb{E}_{\mathbb{P}'} Y$. Then

$$|m - y| + |m - y'| \geq |y - y'| = \Delta,$$

which proves Eq. (7).

Excess Brier loss. For $\mathbb{Q} \in \{\mathbb{P}, \mathbb{P}'\}$, define the best predictor based on L and its risk by

$$\eta_{\mathbb{Q}}(\ell) = \mathbb{E}_{\mathbb{Q}}[Y \mid L = \ell], \qquad R_{\mathbb{Q}}(\phi) = \mathbb{E}_{\mathbb{Q}}[(Y - \phi(L))^2], \qquad R^*_{\mathbb{Q}} = R_{\mathbb{Q}}(\eta_{\mathbb{Q}}).$$

Conditional expectation gives the orthogonal decomposition

$$R_{\mathbb{Q}}(\phi) - R^*_{\mathbb{Q}} = \int (\phi - \eta_{\mathbb{Q}})^2\, d\mu.$$

In particular, the reference risk is the best risk using the same record L, separately in each population. It is not a variance of the residual from an arbitrary predictor. Averaging the two excess risks yields

$$\begin{aligned} &\frac{1}{2} \sum_{\mathbb{Q} \in \{\mathbb{P}, \mathbb{P}'\}} \left(R_{\mathbb{Q}}(\phi) - R^*_{\mathbb{Q}}\right) \\ &\quad = \int \left(\phi - \frac{\eta_{\mathbb{P}} + \eta_{\mathbb{P}'}}{2}\right)^2 d\mu + \frac{1}{4} \int (\eta_{\mathbb{P}} - \eta_{\mathbb{P}'})^2\, d\mu. \end{aligned} \tag{11}$$

The maximum is at least this average. Dropping the first nonnegative term and applying Jensen's inequality gives

$$\max_{\mathbb{Q} \in \{\mathbb{P}, \mathbb{P}'\}} \left(R_{\mathbb{Q}}(\phi) - R^*_{\mathbb{Q}}\right) \geq \frac{1}{4} \int (\eta_{\mathbb{P}} - \eta_{\mathbb{P}'})^2\, d\mu \geq \frac{\Delta^2}{4}. \tag{12}$$

The first bound is attained by $\phi = (\eta_{\mathbb{P}} + \eta_{\mathbb{P}'})/2$. Both bounds are attained when the two conditional success probabilities differ by a constant. The squared-loss result is a standard identity applied to this restricted observation.

A construction on recorded trajectories. Split one domain's training tasks into two disjoint groups, $\mathcal{T}$ and $\mathcal{T}'$. A record ℓ is usable if a successful trajectory in $\mathcal{T}$ and a failed trajectory in $\mathcal{T}'$ both have $L = \ell$. Choose any probability mass function w on the usable records. Under $\mathbb{P}$, draw ℓ from w, then select a successful recorded trajectory with that record from $\mathcal{T}$. Under $\mathbb{P}'$, draw ℓ from the same w, then select a failed recorded trajectory with that record from $\mathcal{T}'$. Both distributions have L-law w. They have success rates one and zero, so $\Delta = 1$ and both best Brier risks are zero. The mean-error and excess-loss lower bounds are 0.5 and 0.25, respectively. The constant predictor $\phi = 0.5$ attains both.

We searched over task splits with greedy local moves, using eight random starts. The selected retail split has 39 and 35 tasks, with 34 usable records. These records occur in 176 of the 1,352 training trajectories (13.0%). The airline split has 18 and 12 tasks, with 7 records occurring in 131 of 539 trajectories (24.3%). Coverage counts all corpus trajectories carrying a usable record. It is not an estimate of how often a deployment must incur the bound. The retail example $[(0, 0), (1, 0), (0, 0)]$ has three assistant turns and occurs 34 times, evenly split between success and failure. It records a noncalling turn, an accepted calling turn, and another noncalling turn.

Selection depends on Y, and the construction reweights the selected runs. Disjoint task sets do not ensure that each task retains its original rollout distribution. The construction therefore establishes existence on the observed support, rather than transfer failure caused solely by changing task sampling. No equal-law claim is made for the unmodified retail and airline corpora. In any single population, $\mathbb{E}[Y \mid L]$ remains a valid conditional success probability.

A.3 First-trigger policies and utility

Fix a baseline policy, a trigger rule, and the continuation policy used after the trigger. For a hidden-risk gate, the first trigger depends only on the history and score available at that turn. Before this trigger the policies coincide; afterward, the verification instruction may affect the rest of the trajectory. Let F be the event that a trigger occurs and let U^g, U^0 be integrable full-trajectory utilities.

Couple the policies so that $U^g = U^0$ on F^c, as in the analysis that reuses untriggered baseline outcomes. For $\mathbb{P}(F) > 0$ and a fixed cost κ per trigger,

$$\begin{aligned}\mathbb{E}[U^g - U^0 - \kappa \mathbb{1}_F] &= \mathbb{E}[\mathbb{1}_F(U^g - U^0)] - \kappa \mathbb{P}(F) \\ &= \mathbb{P}(F)\big(\mathbb{E}[U^g - U^0 \mid F] - \kappa\big).\end{aligned}$$

If $\mathbb{P}(F) = 0$, the gain is zero and no conditional expectation on F is needed. No independence between triggering and utility is assumed. The equation also does not assume that verification affects only the next turn. It separates the probability of intervention from its conditional full-trajectory value.

The identity alone does not estimate that conditional value or identify why an intervention helps or fails. AUROC measures the ordering of local rejection outcomes; it imposes no sign on $U^g - U^0$. For example, if the instruction leaves utility unchanged, the gain is $-\kappa\mathbb{P}(F)$ regardless of the ranking. The empirical estimates compare sampled policy continuations, rather than recovering individual counterfactual effects.

The random timing controls are defined using completed baseline trajectories. Uniform sampling among all baseline turns requires that trajectory's length. The timing-matched control also uses an empirical distribution of baseline trigger positions. They are replay-based controls, rather than fully specified online triggering algorithms. Equation (8) is used only as a policy-value identity and does not imply that these controls can be deployed without modification.

B Experimental details

The subsections below follow the order of Sections 3 to 6.

B.1 Labels

The call indicator J_t is one when the recorded assistant message has a nonempty `tool_calls` field. Ordinary text, including text that resembles a tool call, does not set this indicator. For a calling turn, $Z_t = 1$ when any immediately following tool reply has an error flag. We set $Z_t = 0$ for all other turns. A turn that issues several calls contributes one observation and is rejected when any of its replies is an error. Such turns account for 0.04% of retail calling turns and 0.6% of airline calling turns. Labeling by the first reply instead gives the same label on every turn in all four corpora. We audited rejection labels on the full 14B retail training corpus. There are no orphan tool messages, and 99.96% of calls receive exactly one reply. All $3{,}471$ retail positives begin with the same error prefix and fall into 27 templates, and the top 25 templates cover 99.9% of them. A reverse scan of $9{,}357$ negatives finds no failure wording. A stratified manual check of 53 rows agreed in every case.

A separate audit found 36 turns in the 14B retail training corpus with call JSON in ordinary text and no parsed tool call. The interface never executed these calls, so the recorded labels are $J_t = Z_t = 0$. These turns are excluded from r_2 evaluation and retained as negatives for r_1 and r_3. This follows the interface-based definition of calling; it does not label the model's intended action. We leave the archived labels and results unchanged.

B.2 Token positions

We extract states by replaying each recorded trajectory through the agent's own rendering code, so the token sequence matches the one seen at generation time. Thinking mode is off, so the rendered prompt ends with the assistant header and an empty thinking block. In a calling turn, Qwen3 writes the call as a JSON object with the tool name and its arguments. The object sits between the tags `<tool_call>` and `</tool_call>`, and the message ends with `<|im_end|>`. Table 6 defines the seven readout positions on this format. The example below marks them on the call shown in Figure 1.

```
            <tool_call>[p1] {"name": "get_order_details"[p2],
"arguments": {"order_id":[p3] "W4284542"[p4]}[p5] </tool_call><|im_end|>[p6]
```

Position p_0 lies in the prompt and exists for every assistant turn. Positions p_1 to p_6 exist only for calling turns. In a turn with several calls, p_1 to p_5 refer to the first call. If a position cannot be

Table 6: Readout positions within one assistant turn. For p_1 to p_6, the readout is the last token that ends at or before the stated point in the text.

Position	Readout point	Response content already written
p_0	Last prompt token, after the empty thinking block	None
p_1	End of the opening tag <tool_call>	The decision to call a tool
p_2	End of the tool name value	The chosen tool
p_3	End of the first argument key and its colon	The first argument field
p_4	End of the first argument value	The first argument value
p_5	Last token before </tool_call>	All arguments
p_6	The closing token <|im_end|>	The full call message

found in the generated text, its state is left missing for that turn. The main text uses p_0 for pre-action prediction and p_6 for the comparison after the call is written. The timing analysis below also reads the positions in between.

An earlier version of the extraction reused a key-value cache across turns under a prefix assumption that never held in this template. We corrected this, re-extracted all states, and verified the fix against a no-cache reference, where the median cosine similarity is 0.999997. A simple invariant confirms the fix on every extraction. The number of cache truncations equals the number of turns minus the number of trajectories, since only the first turn of a trajectory has no prefix to reuse.

Timing inside the generated call. Reading later positions raises the score. For the 8B replication, conditional AUROC rises from 0.876 at p_0 to 0.968 once the tool name is complete. The tool name is almost perfectly decodable at that point, with one-versus-rest macro AUROC 1.000. This is consistent with later states holding the realized choice. It does not separate how much of the gain comes from tool identity alone. We use the same decoding as a row-alignment check on held-out data, where it also reaches 1.000 in both domains.

B.3 Comparison predictors

$M1h$ uses eleven observable history features, all of which have values before the action is written. $T1$ and $T2$ fit their vocabularies inside the training folds only. The stack combines base out-of-fold scores, and its second stage is not fully nested with the first. We therefore report the stack as auxiliary and use $T2$ for the primary comparison. The post-action observable model adds the call text, provenance of the argument values, entity-state words near those values, tool identity, and action likelihood. Every one of those features needs the written call, which is why it belongs to the later moment. The text inputs omit the policy text and the tool definitions, which are constant within each domain. Constant additive features cannot change within-domain rankings. Interactions with the varying conversation could matter for richer predictors. Our comparisons therefore concern the stated feature sets and model classes.

B.4 Additional model and probe settings

We also fit probes for Qwen3-8B-AWQ on the same training tasks. Six seeds in retail and eighteen in airline yield 441 and 536 trajectories. Layer 25 and inverse regularization strength 0.01 are fixed from the earlier training-stage analysis. We keep these settings for the local-prediction analyses in both model sizes and fit separate probes for each model and domain. This tests replication across sizes within one family. The independent test evaluation and the comparison after the full call is written cover the 14B model in both domains. The additional target comparison selects regularization within training folds, as described in Appendix B.12.

B.5 Fitting and calibration

Feature scalers and text vocabularies are fitted on training folds and then applied to the corresponding evaluation folds. For the three-target training analysis, each outer fold uses four inner folds grouped by task. The inner predictions supply the scores for fitting the calibration map. The map is then applied to the outer evaluation fold. For independent test evaluation, calibration uses five-fold OOF

scores from all training tasks, and the probe is refitted on all training tasks. The map takes the probe log-odds as its input. The r_3 gate was frozen before test trajectories were generated. The r_1 and r_2 probes were fitted after the intervention outcomes had been examined. Those later fits still used only training data and the previously fixed settings.

B.6 Intervals and analysis status

Ranking intervals use $2{,}000$ bootstrap replicates that resample tasks and keep fitted scores fixed. Paired comparisons use the same resampled tasks and observations for both predictors. These intervals exclude uncertainty from model selection and refitting. Permutation checks use 999 shuffles within task and call-position groups. We refit probes for training-stage checks and keep scores fixed for independent test checks. The model-size replications and intervention designs were written down before their respective runs. Several taxonomy, decomposition, gate-quality, and cost analyses were added later. We identify these as additional analyses where reported. The paired target comparison was designed after the original results and fixed before its first fit. Appendix D.1 reports its implementation and reporting departures.

Metric variants and permutation validity. Besides the pair weighting of Eq. (4) we report a group-macro average, a task-macro average, a coarser position grouping, and a version limited to the first eight call positions. Training-stage permutation checks shuffle labels inside each task and position group and refit the probe on every draw. Held-out checks shuffle labels inside the same groups and keep the frozen scores fixed. That is valid because the frozen probe never sees held-out labels, so refitting would test a different null. The training-stage null medians are 0.500 in both domains, with ranges $[0.472, 0.527]$ in retail and $[0.431, 0.567]$ in airline. On held-out data the medians are 0.500 and 0.501, and the largest draws are 0.555 and 0.643.

B.7 Additional local-prediction results

For the frozen 14B probes, giving each eligible task–position group equal weight yields AUROC 0.907 in retail and 0.847 in airline. Giving each contributing task equal weight yields 0.906 and 0.838. Two coarser call-position groupings also give estimates in the same range. These variants do not include groups with only one label.

The 8B agent has a lower training-archive task pass rate than the 14B agent in both domains. The rates are 21.1% versus 28.0% in retail and 12.1% versus 22.1% in airline. Its positive local-ranking gains in Table 1 therefore also occur for agents with lower task success. The gain intervals overlap across model sizes within each domain, so we do not interpret their point-estimate differences as a model-size effect.

B.8 Comparison after the call is written

At p_6, the full call message has ended and the tool reply is not yet available. This comparison covers the 14B model in both domains and uses training-task OOF predictions. The observable predictor can now read the call text and the features described in Appendix B.3. It reaches matched AUROC 0.961 in retail and 0.956 in airline. The corresponding hidden probes reach 0.969 and 0.951. Their paired gains are $+0.008$ with interval $[-0.002, 0.019]$ and -0.005 with interval $[-0.045, 0.040]$. Figure 2 shows the absolute AUROCs and paired gains at both observation times. Appendix B.2 reports the intermediate readouts and the tool-identity alignment check.

B.9 Rejection types

This auxiliary analysis concerns 14B local rejection prediction on training tasks. We classify rejections using an ordered list of explicit patterns, with complete coverage and no unmatched string in either domain. The classes are referential, entity state, argument internal, policy only, and dispatcher. A referential rejection concerns whether an identifier exists. An entity-state rejection requires the current state of an entity together with a business rule. An argument-internal rejection can be decided from the call's own arguments. A policy-only rejection needs a rule but no database read. Retail has 91 argument-internal rejections and no policy-only ones; airline has 42 policy-only rejections and no argument-internal ones.

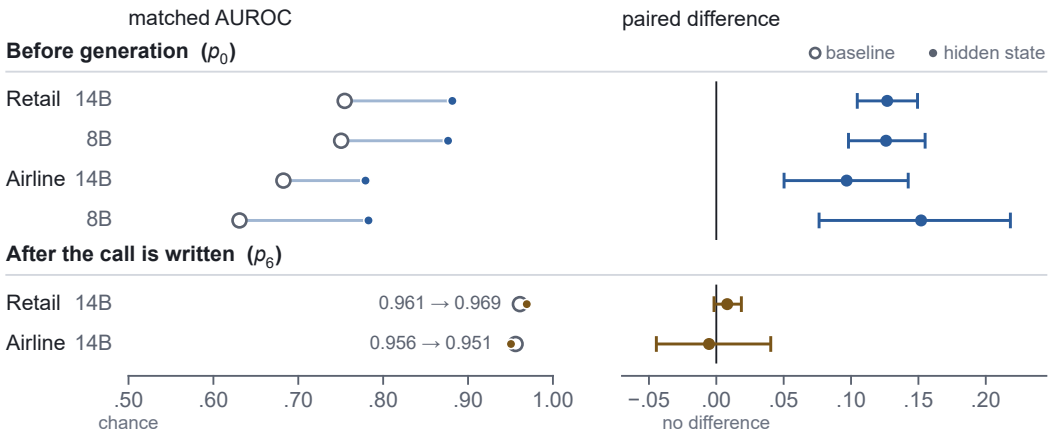

Figure 2: Tool-rejection ranking before generation (p_0) and after the call is written (p_6). Left: baseline (open) and hidden-probe (filled) AUROC. Right: paired hidden-minus-baseline differences with 95% task-bootstrap intervals. The baselines use pre-action text at p_0 and written-call observables at p_6. Scores are out-of-fold on training tasks. AUROCs within task and exact call position are weighted by eligible pairs of rejected and accepted calls. Only the 14B model is evaluated at p_6. Differences between gains at the two times were not tested.

Retail rejections are 78.6% referential and 18.4% entity-state, compared with 42.1% and 50.1% in airline. The pre-action probe's referential AUROC is 0.905 in retail and 0.853 in airline. Its entity-state AUROC is 0.797 and 0.713. The paired gaps are 0.108 $[0.070, 0.147]$ and 0.139 $[0.036, 0.223]$. We pre-registered an ordering across all five classes, and it did not hold. The referential-versus-entity-state contrast is the only cross-type finding supported in both domains. Held-out groups are too small to establish replication on unseen tasks.

The pooled cross-domain AUROC gap of 0.102 decomposes into a composition term of 0.035 $[0.007, 0.068]$ and a within-type term of 0.067 $[0.017, 0.124]$. The decomposition is an exact weighted identity, with a numerical residual at machine precision. It describes the observed gap; it does not identify two causal mechanisms. This analysis distinguishes local error types and does not by itself establish a limit on task-success prediction.

B.10 Calibration diagnostics

For a probability score p and binary outcome B, we fit the diagnostic model

$$\mathbb{P}(B = 1 \mid p) \approx \sigma(\alpha + \beta \operatorname{logit}(p)), \qquad \operatorname{logit}(p) = \log \frac{p}{1-p},$$

where $\sigma(x) = 1/(1 + \exp(-x))$. The intercept α and slope β are fitted jointly on evaluation outcomes. They diagnose the evaluated scores; they are not used to modify them. The reference pair is $(0, 1)$. A slope below one suggests overly spread log-odds, while a nonzero intercept allows an offset. These diagnostics do not fully specify the calibration curve (Van Calster et al., 2019). In particular, an intercept fitted jointly with a slope is not the same statistic as an intercept fitted with slope fixed at one.

For held-out ECE, we sort the scores and split observations into ten bins of nearly equal size. We average each bin's absolute difference between its mean score and event rate, weighted by bin size. Table 2 reports both raw and calibrated ECE values. Its AUROC is pooled over each target's eligible turns, whereas Section 4.1 matches task and call position. The targets have different labels and eligible sets, so their AUROCs are not a ranking of target difficulty. Scores are clipped to $[10^{-6}, 1 - 10^{-6}]$ for log-odds and log-loss calculations. The repeated-sampling diagnostics instead fit the logistic model to the sampled successes and failures, using the weights in Appendix B.11. Their intercepts and slopes are therefore separate from the held-out diagnostics.

Table 7: Eligible prefixes and actual draws in each repeated-sampling analysis. The q_2 rows require at least one sampled call and a stored r_2 score. Other rows use all responses from the panel. The last column gives draws per eligible prefix.

Panel	Targets	Prefixes	Draws	Per prefix
Retail, calling turns	q_1, q_3	500	8,000	16
	q_2	498	7,702	1–16
Airline, calling turns	q_1, q_3	500	8,000	16
	q_2	495	7,632	2–16
Retail, all turns	q_1, q_3	500	8,000	16
	q_2	338	5,260	2–16

B.11 Repeated-prefix sampling and outcome definitions

All three panels use 14B training trajectories and sample 16 responses per prefix. The probe predictions are OOF scores, with calibration fitted inside the training folds as described in Appendix B.5. The new sampled outcomes do not fit either the probe or the calibration maps.

The sampling implementation executes only the first parsed tool call in each response. Let $B = 1$ if that call is rejected, and set $B = 0$ if no call occurs. For these panels we define

$$q_1(c, s) = \mathbb{P}(J = 1 \mid c, s), \quad q_2(c, s) = \mathbb{P}(B = 1 \mid J = 1, c, s), \quad q_3(c, s) = \mathbb{P}(J = 1, B = 1 \mid c, s).$$

Here conditioning on (c, s) means restoring both the visible context and environment state. The original trajectory target Z instead checks whether any call in the turn is rejected. The two rejection labels agree for single-call responses. The sampling archive does not retain additional calls, so agreement for every sampled response cannot be verified. We therefore treat these panels as a first-call audit of the probes. A score difference can reflect this outcome-definition difference as well as the change in evaluation population.

Every panel contains 8,000 recorded responses, all marked as either a call or text. There are no recorded malformed responses or execution exceptions. The retail and airline calling-turn panels contain 7,702 and 7,632 sampled calls. The all-turn panel contains 5,402 calls and 2,598 text responses. State restoration was checked separately by replaying 250 recorded calls and obtaining identical tool replies.

Sampling frames and weights. The calling-turn frames require a recorded call, a stored score, and at least one earlier call. Their sizes are 11,476 prefixes in retail and 3,263 in airline. The retail panel covers 500 of 580 strata and the airline panel covers all 210 strata. The all-turn retail frame contains 21,455 assistant turns, of which 40.2% are text-only. Its panel covers 500 of 594 strata. Strata combine task, position bucket, and score tercile. The position buckets are indices 0–1, 2–7, and 8 onward. Position means call index in the calling-turn frames and assistant-turn index in the all-turn frame.

The sampler shuffles prefixes within strata, shuffles the strata, and visits them in round-robin order. For a sampled prefix in stratum h, the analysis uses weight $w_i = N_h/n_h$. Here N_h is the frame stratum size and n_h is the number sampled from it. The two retail panels have one prefix per sampled stratum; the airline panel has one to three. Weights are normalized over the eligible prefixes for each reported statistic. The sums of stratum weights before outcome-specific exclusions are 9,834, 3,263, and 18,368, respectively. The corresponding effective sample sizes, $(\sum_i w_i)^2 / \sum_i w_i^2$, are 281.4, 211.8, and 324.9. We report weighted panel estimates and retain the stratum coverage as a limitation.

In the all-turn panel, 365 prefixes produce at least one sampled call. However, the score archive stores r_2 only for originally recorded calling turns. Only 339 of the selected prefixes have that score, and one of them produces no sampled call. The q_2 score analysis therefore uses 338 prefixes and 5,260 calls. Its support is selected by both original score availability and the new call samples. It does not evaluate r_2 over the full all-turn frame. The calling-turn panels also cannot evaluate r_1 over an all-turn population, since their frames were selected on the original call decision.

Losses and finite-sample correction. For one target, let K_i be its eligible draw count, k_i its positive count, and $\widehat{q}_i = k_i/K_i$. For q_1 and q_3, $K_i = 16$. For q_2, K_i is the number of sampled calls

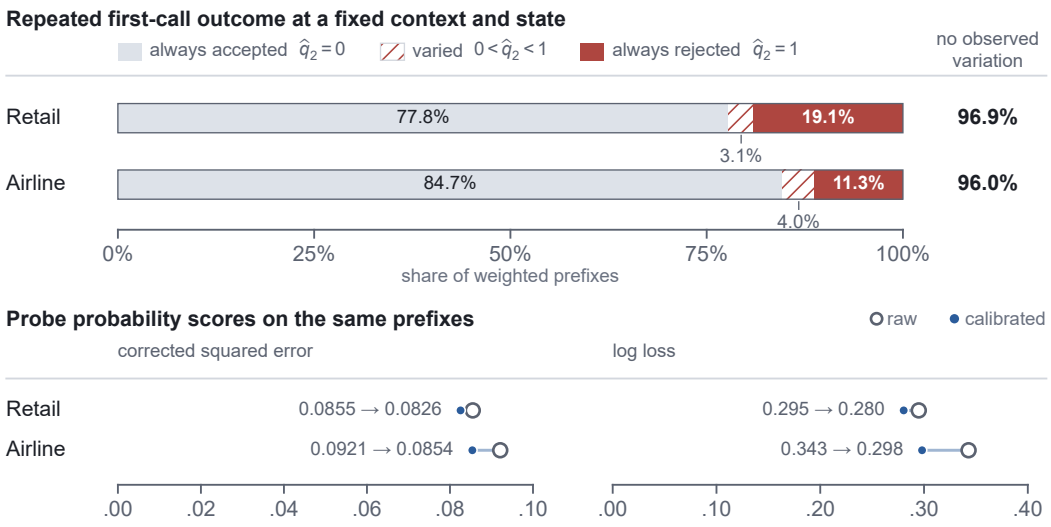

Figure 3: Repeated first-call outcomes and probe probabilities on 14B training prefixes. Each domain samples 16 responses at each of 500 restored contexts and states; 498 retail and 495 airline prefixes have eligible calls. Top: weighted shares by empirical first-call rejection frequency $\widehat{q}_2$, rounded to sum to 100%. No observed variation does not imply a true probability of zero or one. Bottom: raw and calibrated OOF probability scores, shown as descriptive point estimates. Prefix shares and squared error use stratum weights; log loss also weights by sampled call counts. The squared-error correction is omitted for two retail prefixes with only one sampled call. This audits the first call, rather than all calls in a turn.

and can be smaller. Let p_i be the corresponding probe probability. The reported log loss is

$$-\frac{\sum_i w_i\{k_i \log p_i + (K_i - k_i)\log(1 - p_i)\}}{\sum_i w_i K_i}.$$

It gives weight proportional to $w_i K_i$ to each prefix's average loss. The corrected squared error uses prefix weights:

$$\widehat{D} = \frac{\sum_i w_i\{(p_i - \widehat{q}_i)^2 - \widehat{q}_i(1 - \widehat{q}_i)/(K_i - 1)\}}{\sum_i w_i}, \qquad K_i > 1.$$

Thus, the two scores use different prefix weights when call counts differ.

Conditional on (C_i, S_i) and K_i, independent draws give $k_i \sim \text{Binomial}(K_i, q_i)$. Then

$$\mathbb{E}[(p_i - \widehat{q}_i)^2] = (p_i - q_i)^2 + \frac{q_i(1 - q_i)}{K_i}, \quad \mathbb{E}\left[\frac{\widehat{q}_i(1 - \widehat{q}_i)}{K_i - 1}\right] = \frac{q_i(1 - q_i)}{K_i}.$$

Their difference estimates squared error against the fixed-state probability, rather than Brier loss for a single draw. Two eligible retail calling-turn prefixes have $K_i = 1$. For these, the archived analysis sets the correction to zero. This convention retains their observations but prevents an exactly unbiased interpretation of the aggregate. Prefixes with no sampled call are excluded from q_2. The Bernoulli-variance estimate uses $K_i\widehat{q}_i(1 - \widehat{q}_i)/(K_i - 1)$ for $K_i > 1$ and zero for $K_i = 1$. It has the same singleton limitation. Finite-sample corrected squared-error estimates can be negative, even though their population targets are nonnegative. Figure 3 summarizes first-call rejection frequencies and probability scores in the two calling-turn panels.

The diagnostic logistic fits weight each sampled success and failure by w_i. Consequently, their effective prefix weights are also proportional to $w_i K_i$. The calibration maps themselves were fitted on the original training folds, not on these new weighted outcomes. For example, the all-turn panel's rejection analysis has a calibrated slope of 1.000 but a diagnostic intercept of $+0.442$. The reference intercept is zero, so a slope near one can coexist with an offset.

Table 8: Probability scores and logistic diagnostics in the repeated-sampling panels. Arrows show raw $\rightarrow$ calibrated scores. The last two columns give the jointly fitted intercept and slope (α, β). Loss comparisons are descriptive point estimates. The q_2 rows use the restricted sets in Table 7.

Panel / target	Log loss raw $\rightarrow$ cal.	Corrected SE raw $\rightarrow$ cal.	(α, β) raw	(α, β) calibrated
Retail calls, q_2	$0.295 \rightarrow 0.280$	$0.0855 \rightarrow 0.0826$	$(-0.395, 0.656)$	$(-0.154, 1.081)$
Retail calls, q_3	$0.303 \rightarrow 0.286$	$0.0813 \rightarrow 0.0786$	$(-0.283, 0.660)$	$(-0.130, 0.965)$
Airline calls, q_2	$0.343 \rightarrow 0.298$	$0.0921 \rightarrow 0.0854$	$(-0.456, 0.477)$	$(-0.385, 0.830)$
Airline calls, q_3	$0.329 \rightarrow 0.279$	$0.0754 \rightarrow 0.0737$	$(-0.289, 0.501)$	$(-0.173, 0.828)$
Retail all, q_1	$0.173 \rightarrow 0.165$	$0.0313 \rightarrow 0.0291$	$(0.254, 0.728)$	$(0.239, 0.973)$
Retail all, q_2	$0.404 \rightarrow 0.364$	$0.1092 \rightarrow 0.1063$	$(0.215, 0.605)$	$(0.442, 1.000)$
Retail all, q_3	$0.267 \rightarrow 0.249$	$0.0711 \rightarrow 0.0690$	$(0.220, 0.692)$	$(0.371, 1.005)$

Observed variation and its scope. No rejection variation means $\widehat{q}_2 \in \{0, 1\}$ among the sampled calls. Its weighted frequency is 0.9688 in retail and 0.9596 in airline for the calling-turn panels. The corresponding Bernoulli-variance estimates are 0.00591 and 0.00798. Tool choice is unchanged on 94.1% and 92.1% of prefixes with sampled calls. These summaries concern a finite number of samples at fixed states. Even with 16 calls, a rejection probability of 0.05 produces no rejection with probability $0.95^{16} \approx 0.44$. No observed variation therefore leaves room for rare errors.

In the all-turn panel, the weighted frequencies of no observed variation for q_1, q_2, and q_3 are 0.8648, 0.9673, and 0.9592. Their Bernoulli-variance estimates are 0.02413, 0.00668, and 0.00758. The q_2 support differs from the other two, and Bernoulli variance also depends on the event rate. We therefore do not interpret their ratios as an ordering of uncertainty across targets. Nor do we compare rejection rates across the two retail sampling frames as a change in agent behavior. The two panel extensions were specified before their runs; the theoretical decomposition and the present scope audit were added later.

B.12 Paired local and task targets

We compare two separately fitted probes on the same calling-turn prefixes. The local label is turn-level rejection Z_t; the task label is final failure $1 - Y$. We use the 14B training archives only. No new trajectories or independent test predictions are used. The design was fixed before fitting, after the original paper results were known. Retail is the primary analysis. Airline was designated descriptive before fitting because the shared support contains few tasks.

Each hidden probe reads the $5{,}120$-dimensional layer-25 vector at p_0. This layer was selected in the earlier rejection analysis. Linear probes test what a specified reader can extract from a representation (Alain & Bengio, 2016). Probe capacity and controls affect that interpretation (Hewitt & Liang, 2019). Our two targets share a reader family and tuning budget; this is not a control-task selectivity experiment.

The text baseline uses the final 800 characters of the preceding conversation. It fits character 3–5-gram TF–IDF features within word boundaries. The minimum document count is 3, the vocabulary cap is $200{,}000$, and term frequency is log-scaled. The history baseline uses eleven values. They include previous calling turns, message index, previous rejections, prompt length, and distance from the last accepted tool reply. The remaining values are previous tool-call count, user-message count, text-turn count, last observation length, last error flag, and distinct tool count. No input uses the current response, its tool reply, final trajectory length, or evaluator information.

Both targets and all predictors share five outer folds grouped by task. Four inner task-grouped folds select inverse regularization strength C by trajectory-weighted log loss. The grids are $\{0.001, 0.01, 0.1, 1\}$ for hidden and history features and $\{0.3, 3, 30, 300\}$ for text. Within a feature family, both targets use the same grid. Ties select the smaller C. Each trajectory has equal total weight in classifier fitting and validation loss. Within each training split, its prefix weights are inverse to its number of eligible prefixes, then normalized to mean one. The logistic solver uses at most $4{,}000$ iterations and no class weights. Scalers and text vocabularies are fitted only on the

Table 9: Support for the paired target comparison. Shared groups contain both label values for both targets.

Quantity	**Retail**	**Airline**
Eligible prefixes	12,828	3,802
Trajectories / tasks	1,352/74	539/30
All task–position groups	1,200	423
Groups with both rejection labels	717	140
Groups with both final-failure labels	617	109
Shared groups / tasks	430/55	41/12
Shared prefixes / trajectories	6,092/999	501/215
Prefix coverage	47.5%	13.2%
Rejection pairs in shared groups	14,775	1,325
Final-failure pairs in shared groups	15,150	1,352

corresponding training folds. Scaling is unweighted in the implementation; Appendix D.1 records the departure from the planned weighting.

Calling prefixes join hidden indices and trajectory outcomes by trajectory identifier and message index. The archived reward is classified as success at 0.999 or above. Both labels are available for all $12{,}828$ retail and $3{,}802$ airline calling prefixes. There are 5 retail and 22 airline turns with multiple calls. The local label checks all recorded tool replies for the turn. The observed prefix-level rejection rates are 0.2706 and 0.1623. The corresponding final-failure label rates are 0.7658 and 0.8411. These are unweighted prefix frequencies, not trajectory failure rates.

We train on all eligible prefixes and restrict matched evaluation to shared groups. Each group fixes the task and exact call position and contains both label values for both targets. Let $\mathcal{T}_*$ be the contributing tasks and $\mathcal{G}_i$ the shared groups for task i. For predictor M and target $b \in \{Z, 1-Y\}$, we report

$$A_{M,b} = \frac{1}{|\mathcal{T}_*|} \sum_{i \in \mathcal{T}_*} \frac{1}{|\mathcal{G}_i|} \sum_{g \in \mathcal{G}_i} \mathrm{AUC}_g(s_{M,b}). \tag{13}$$

Within-group AUROC gives half credit to ties. The groups and averaging weights are identical across targets and predictors. This differs from Eq. (4), whose pair weights depend on each target's label counts. With H denoting the hidden probe and O the text baseline, the contrasts are

$$D_H = A_{H,Z} - A_{H,1-Y}, \qquad G_b = A_{H,b} - A_{O,b}, \qquad D_G = G_Z - G_{1-Y}.$$

The primary contrast is D_H; the others compare gains over text. The design specified 0.10 as a practically relevant D_H and did not specify an equivalence test.

Table 9 separates prefix, trajectory, and task counts. A trajectory contributes at most one prefix to any exact-position group, but may appear in several groups. Its final label is therefore repeated across positions. Local outcomes within a trajectory can also be correlated. Neither the prefix count nor the trajectory count is a count of independent task replicates. Shared support selects task positions with variation in both labels, so the estimates do not cover all calling prefixes.

We resample contributing tasks $2{,}000$ times with seed 20260916. Every contrast uses the same task draws and fixed OOF predictions. Each drawn task retains its within-task group average; repeated draws count as separate copies. The resulting intervals exclude uncertainty from refitting and parameter selection. The overlapping OOF training folds are also not resampled. Figure 4 shows the shared-group scores and paired contrasts. Table 10 gives all fitted predictors and their intervals.

In retail, final-failure AUROC is 0.549 $[0.510, 0.586]$. This indicates some task-related ranking under the specified evaluation, despite the larger local score. Its gain over text is not detected. The direct gain difference is positive in retail, but its airline interval includes zero (Table 3). The latter cannot be inferred from one gain excluding zero and the other including it.

Table 11 reports pooled AUROC on all eligible calling prefixes, with equal total row weight per trajectory. The hidden final-failure scores are higher than the matched scores. The comparison changes task and position comparisons, eligible support, and weights together. It does not isolate task difficulty or show that most predictive performance comes from identifying difficult tasks.

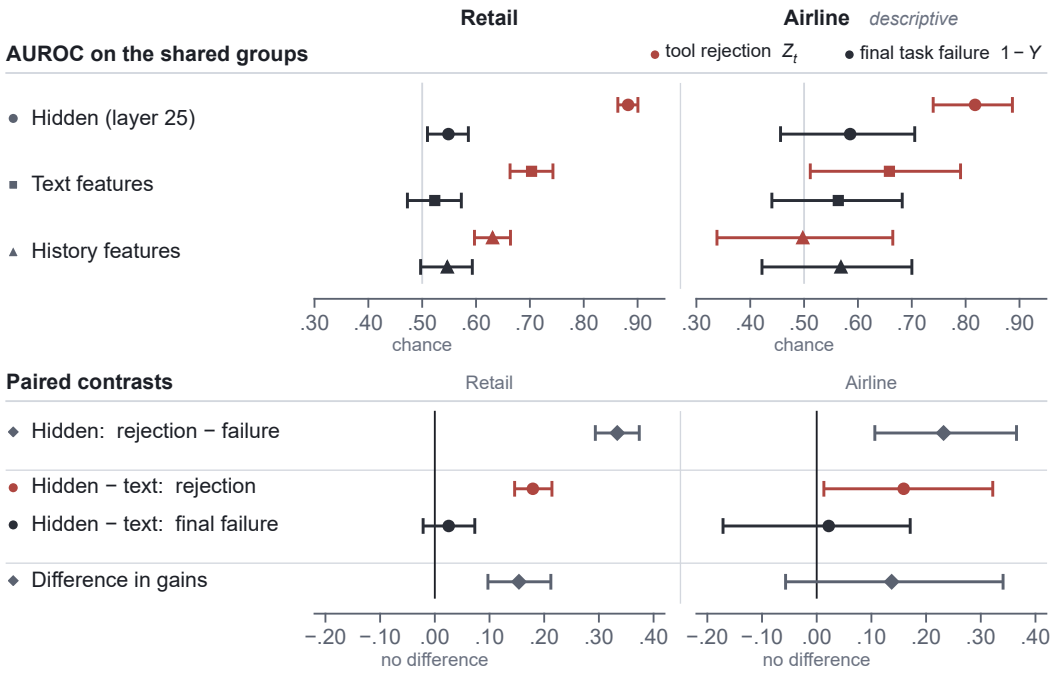

Figure 4: Tool rejection and final task failure at p_0. All results use task-grouped out-of-fold predictions on Qwen3-14B training trajectories. Predictors are fitted separately for each target; hidden probes use layer 25. Top: AUROC on shared task and exact-call-position groups where both targets vary. Group AUROCs are averaged equally within each task, then equally across tasks. Bottom: paired contrasts; the last row subtracts the final-failure hidden-over-text gain from its rejection gain. Bars show 95% task-bootstrap intervals with scores held fixed. Shared groups cover 47.5% of calling prefixes in retail and 13.2% in airline. Airline is descriptive, with 12 contributing tasks.

Table 10: Matched AUROC on shared groups, with 95% task-bootstrap intervals. Airline is a descriptive comparison.

Domain	**Predictor**	**Rejection** Z	**Final failure** $1 - Y$
Retail	Hidden	$0.882\ [0.863, 0.901]$	$0.549\ [0.510, 0.586]$
	Text	$0.703\ [0.663, 0.743]$	$0.523\ [0.473, 0.573]$
	History	$0.631\ [0.597, 0.664]$	$0.547\ [0.497, 0.593]$
Airline	Hidden	$0.818\ [0.740, 0.887]$	$0.586\ [0.456, 0.705]$
	Text	$0.658\ [0.512, 0.791]$	$0.564\ [0.440, 0.682]$
	History	$0.498\ [0.338, 0.665]$	$0.568\ [0.422, 0.700]$

Table 12 gives all four position groups specified before fitting. Retail's later groups have higher final-failure point estimates. The groups contain different tasks and trajectories, and no trend interval was computed. They do not establish a within-trajectory increase in readability. The last airline group contains only three tasks.

Every hidden fit in both domains and both targets selects $C = 0.001$, the strongest penalty in the grid. Text selects $C = 0.3$ except in one retail rejection fold, which selects 3. The grid was not expanded after observing the results. Thus, the study compares the specified fitting procedures and does not establish each representation's best possible performance. It also leaves symmetric layer selection, nonlinear readers, and independent new-task confirmation untested. The final outcome depends on future interactions as well as the current prefix. The observed gap does not separate these sources of difficulty.

Table 11: Pooled AUROC on all eligible calling prefixes, using trajectory weights. These scores use a different support and weighting from the matched comparison.

Predictor	Retail		Airline	
	Z	$1-Y$	Z	$1-Y$
Hidden	0.917	0.613	0.819	0.721
Text	0.794	0.574	0.681	0.592
History	0.764	0.562	0.598	0.523

Table 12: Hidden-probe point estimates by call position. Group and task counts refer to shared support within each position range.

Domain	Call index	Groups/tasks	$A_{H,Z}$	$A_{H,1-Y}$	D_H
Retail	0	41/41	0.909	0.416	+0.493
	1–2	80/49	0.939	0.518	+0.421
	3–7	172/50	0.870	0.586	+0.284
	8+	137/40	0.843	0.636	+0.207
Airline	0	8/8	0.829	0.637	+0.192
	1–2	11/6	0.805	0.592	+0.213
	3–7	14/6	0.816	0.531	+0.286
	8+	8/3	0.956	0.292	+0.664

For the local hidden probe, restricting to shared groups changes the task-equal AUROC from 0.8810 to 0.8824 in retail. The corresponding airline change is from 0.8082 to 0.8175. This check finds little change in local ranking, but does not rule out selection effects on the final-failure target or the gap. Under the original pair-weighted metric, the new local fits score 0.8770 and 0.7662. Their differences from the earlier archived scores are −0.0042 and −0.0133. The fitting weights and regularization selection differ, so this is a consistency check rather than an exact replication. Constant scores give AUROC 0.5 for both targets and gap zero. Random-score diagnostics are estimator checks, not lower bounds on sampling uncertainty or formal power calculations.

B.13 Local execution and task-success comparisons

These analyses use 14B training runs from 74 retail tasks and 30 airline tasks. For each task with both clean and nonclean trajectories, we compute the difference in their empirical success rates. The within-task estimate averages these differences equally across 49 eligible retail tasks and 20 airline tasks. For the write-count comparison, we form groups by task and realized write-call count. There are 58 eligible retail groups and 34 airline groups. Each eligible group again receives equal weight. For both estimates, the 2,000 bootstrap draws resample the original tasks. Repeated copies of a task remain separate clusters.

The activity counter classifies each calling turn using its first parsed tool call. Names beginning with `get_`, `find_`, `list_`, `calculate`, `think`, or `transfer_` are treated as non-writing. Other names count as a write attempt, including attempts that are rejected. This is a measure of recorded activity, not successful state change. An audit found 5 multi-call turns in retail and 22 in airline. The rejection labels and full binary records are identical whether we check the first call or all immediate tool replies on these archived runs. The earlier record construction therefore uses the same L as Section 5.2.

The zero-rejection group contains 284 trajectories from 50 retail tasks, with 127 successes. It contains 290 trajectories from 25 airline tasks, with 41 successes. The trajectory-pooled success rates are 0.447 $[0.308, 0.586]$ and 0.141 $[0.030, 0.296]$. For these intervals, tasks with at least one clean trajectory are sampled uniformly with replacement within each domain. Each selected task contributes all its clean trajectories. The replicate rate is the pooled success count divided by the pooled trajectory count, rather than an equal-weight average of task rates.

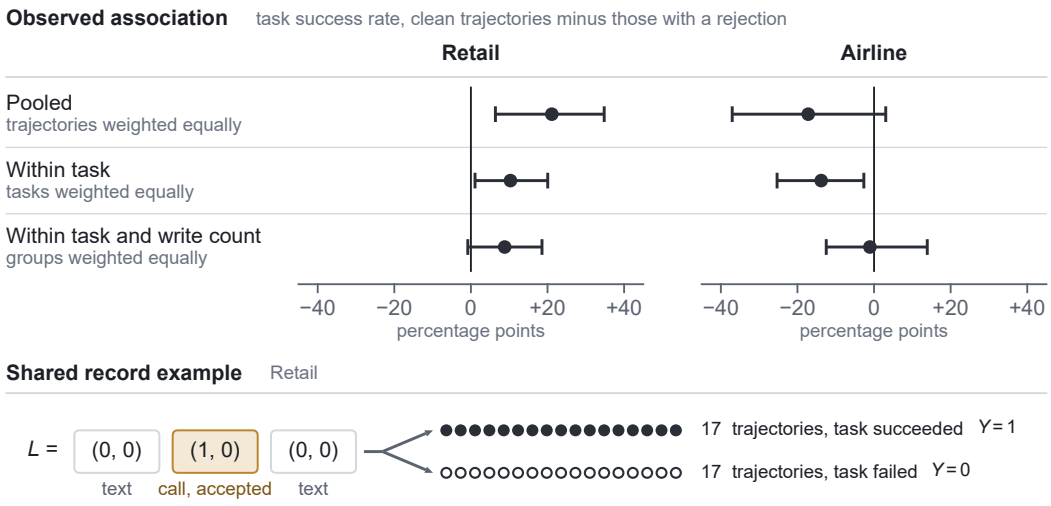

Figure 5: Local execution records and task outcomes on 14B training trajectories. Top: clean-minus-nonclean differences in success rate, with 95% task-bootstrap intervals. Clean means no tool rejection. The rows use different eligible groups and weights and do not estimate causal effects. Bottom: 17 successful and 17 failed retail trajectories share the same binary record L. Entries give (J_t, Z_t); text denotes a noncalling turn. Mixed outcomes alone do not rule out probability prediction. The transfer bound in Proposition 3 also requires two populations with the same distribution of L, as constructed in Appendix A.2.

We added the direct retail-minus-airline difference after reviewing these results. Using 2,000 independent domain-wise cluster draws gives 0.306 $[0.106, 0.486]$. This post-hoc analysis uses the same zero-rejection groups and reproduces their archived marginal intervals. The domains have no paired tasks. No new trajectories are generated and no probe is fitted. The construction search and this interval are evidence on the training corpora, not held-out transfer tests. Figure 5 summarizes the association estimates and illustrates the shared retail record.

B.14 Reference alignment and terminal probes

This retail analysis requires both evaluator action checks and a stored terminal vector. There are 1,260 such trajectories from 72 tasks. Requiring at least one reference write leaves 1,187 trajectories from 68 tasks. The evaluator records whether each reference action was matched, including its arguments. The argument-level oracle uses four features: the fraction of reference writes matched, whether all were matched, their count, and the fraction of all reference actions matched. The name-level oracle instead compares the multiplicities of performed and required write-tool names. It uses the matched fraction, the all-matched indicator, and the required count. Thus the two oracles differ in feature content as well as matching resolution.

Both predictors fit standardized logistic regression with $C = 0.01$ using five task-grouped folds. Their terminal-failure AUROCs use positive–negative trajectory pairs within the same task, weighted by the number of such pairs. This differs from the task-and-call-position comparison for the turn-level probe. There are 51 eligible tasks and 2,670 pairs for terminal failure. The argument-level oracle reaches 0.913 $[0.871, 0.949]$ and the name-level oracle reaches 0.667. The oracle uses evaluator information related to the task-success criteria. Its performance is neither a deployable baseline nor evidence independent of that definition.

We also reassign reference actions across tasks 1,000 times, allowing no task to keep its own reference. Each draw recomputes the name-level features and refits the task-grouped predictor. The mean AUROC is 0.502 and the maximum is 0.582. None reaches the true-reference score, giving a Monte Carlo tail fraction $(0 + 1)/(1000 + 1) = 0.001$. Correct reference assignment therefore performs better than these mismatched-reference controls.

The terminal probe reads layer 20 at the end of the last recorded assistant response. Its target is whether all reference writes were matched. The observable comparator uses message count, rejection count, write-call count, and the number of distinct write tools. Both use the same standardized logistic regression and task-grouped folds. For this target, 54 tasks contribute $2{,}961$ within-task pairs. The hidden and observable scores are 0.675 and 0.588, with paired difference 0.087 $[0.009, 0.170]$. The interval resamples tasks while holding OOF predictions fixed.

A smaller analysis restricts this sample to 250 trajectories with no tool errors, from 46 tasks. The terminal-failure probe then reaches 0.491 $[0.390, 0.629]$, based on 21 tasks containing both outcomes. This wide interval does not show that task-success information is absent. These terminal probes use a later observation time and a selected population, so they are not matched comparisons with the pre-action rejection probe.

B.15 Intervention protocol and pairing

The hidden gate reads the calibrated r_3 score at p_0. The hidden and observable thresholds were fixed at their respective 90th percentiles on training OOF scores. Neither threshold was adjusted for the held-out intervention outcomes. The observable gate combines text predictors and visible history features; it does not read hidden states.

The original study contains arms $A1$–$A6$. $A1$ is the original agent, and $A2$ applies verification under the hidden gate. The random control $A3$ attempts to trigger with the hidden gate's observed trajectory-level firing probability, then chooses a baseline turn uniformly. $A4$ applies verification under the observable gate. $A5$ applies verification from the first generated assistant turn. $A6$ branches at the same points as $A2$ but keeps the original instruction. The additional timing control $A7$ is described in Appendix B.16. The model weights and tools remain fixed across arms.

The verification instruction is:

> Before acting, verify with a read-only lookup that the entities you are about to reference exist and are in a state that permits this operation.

It is appended to the agent policy and remains active for the rest of the trajectory. This persistent delivery replaced the initially proposed single-message injection because of the interface constraints. The change was recorded before inspecting test outcomes. There is no second trigger or adaptive change to the instruction.

Each branch restores the baseline message history and environment state immediately before its trigger. Replaying 250 recorded calls reproduced all tool replies in a separate restoration check. The agent and simulated user then sample new continuations at temperature 0.7, with an 80-step limit. Continuation seeds depend on both the arm and the baseline trajectory. Thus, pairing means a shared baseline trajectory, not identical post-trigger random draws.

There are eight baseline runs for each of 40 retail tasks and 20 airline tasks. All arms cover the same 320 and 160 baseline trajectories. The primary analysis includes trajectories that never trigger, with their baseline outcome unchanged. It measures the average effect of applying the full trigger policy on this population. It does not restrict evaluation to successfully completed branches.

For each contrast, the bootstrap resamples tasks with replacement and carries all paired trajectory differences from the selected tasks. There are $2{,}000$ replicates. The random seed for each primary interval is determined by its input differences and task identifiers, so adding another contrast does not change that interval. The pooled success comparison resamples the 60 domain-qualified task clusters. Its point estimate weights retail and airline by their trajectory counts, $2/3$ and $1/3$. It summarizes this mixture and does not assert a common treatment effect across domains. The pooled $A2 - A1$ success gain is 2.9 $[0.2, 5.8]$ percentage points.

The main endpoint counts turns with any immediate tool error, rather than individual errors within a multi-call turn. Lower counts favor the intervention. Early protocol tables retained the opposite sign convention from the original success endpoint. We report all contrasts as intervention minus comparator and state the direction for each endpoint. Correcting that sign wording changes no decision here, since the relevant rejection intervals include zero.

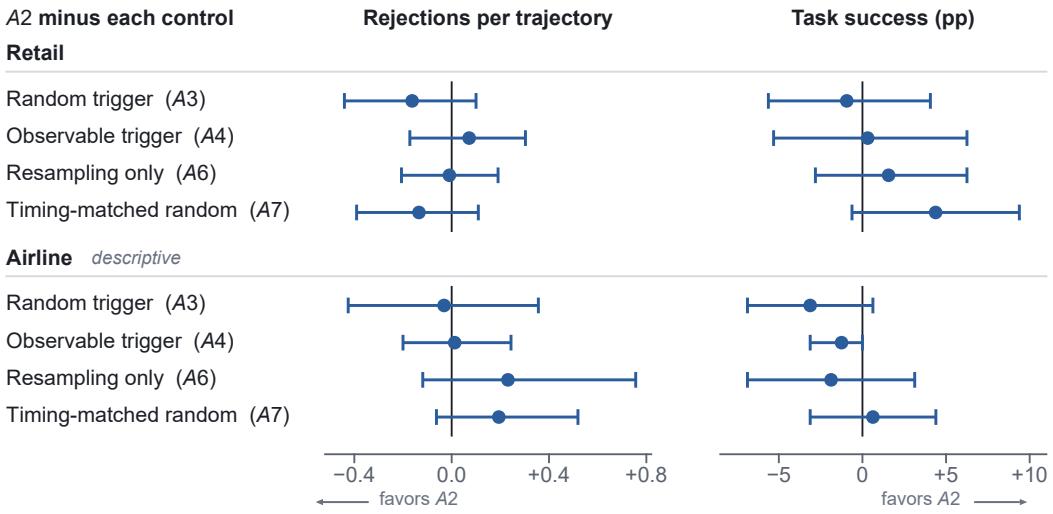

Figure 6: Hidden-triggered verification ($A2$) versus each control on held-out 14B trajectories. Differences average over all 320 retail trajectories from 40 tasks and 160 airline trajectories from 20 tasks. Nontriggered trajectories keep their original outcomes; failed branches use baseline outcomes. Bars show paired 95% task-bootstrap intervals. Negative rejection differences and positive success differences favor $A2$; pp denotes percentage points. Airline results are descriptive. All intervals include zero, which does not establish equivalence.

Table 13: All verification arms relative to the original agent. Fire counts are attempted triggers; unsuccessful branches are included in those counts. The main analysis substitutes baseline outcomes for failed branches. Rejection and success differences have 95% task-cluster bootstrap intervals. Token differences are mean completion-token counts per trajectory.

Domain	Arm	Fire	Δ rejections	Δ success (pp)	Δ tokens
Retail	$A2$	208/320	-0.013 $[-0.259, 0.266]$	$+4.7$ $[0.9, 8.8]$	-18.0
	$A3$	194/320	$+0.150$ $[-0.006, 0.309]$	$+5.6$ $[1.9, 9.4]$	$+78.2$
	$A4$	294/320	-0.084 $[-0.316, 0.138]$	$+4.4$ $[-0.6, 9.7]$	$+1.7$
	$A5$	320/320	-0.019 $[-0.309, 0.281]$	$+1.6$ $[-2.8, 6.3]$	$+30.5$
	$A6$	208/320	-0.003 $[-0.222, 0.238]$	$+3.1$ $[-1.3, 7.2]$	$+97.0$
	$A7$	194/320	$+0.122$ $[-0.044, 0.325]$	$+0.3$ $[-2.8, 3.4]$	$+4.2$
Airline	$A2$	93/160	$+0.131$ $[-0.156, 0.500]$	-0.6 $[-3.1, 1.3]$	$+5.8$
	$A3$	88/160	$+0.163$ $[-0.038, 0.431]$	$+2.5$ $[-0.6, 5.0]$	$+85.1$
	$A4$	88/160	$+0.119$ $[-0.013, 0.281]$	$+0.6$ $[-1.9, 3.8]$	-5.9
	$A5$	160/160	$+0.138$ $[-0.025, 0.331]$	$+0.6$ $[-3.8, 5.0]$	$+77.4$
	$A6$	93/160	-0.100 $[-0.319, 0.063]$	$+1.3$ $[-3.1, 6.3]$	-80.3
	$A7$	88/160	-0.063 $[-0.231, 0.094]$	-1.3 $[-4.4, 1.3]$	$+29.6$

The retail baseline success rate is 28.1% and its rejection count is 2.288 per trajectory. The airline values are 10.6% and 1.019. Table 13 gives each arm relative to the baseline. Table 14 gives the airline direct comparisons. The latter remain descriptive, as specified before the original runs. Figure 6 shows the direct contrasts in both domains.

Failed branches. A trigger attempt can fail to produce a completed simulation. This happened in two airline $A2$ branches, one airline $A6$ branch, and one retail $A7$ branch, all after exceeding the context limit. Attempted triggers are therefore distinct from completed branches. For example, airline $A2$ attempts 93 branches but completes 91. The firing counts in Table 13 include the attempts.

The main analysis substitutes the corresponding baseline outcomes for these failures. This convention was chosen after failures were encountered in the original study. We also report post-hoc

Table 14: Descriptive direct comparisons on 160 held-out airline trajectories. Each difference is $A2$ minus the named control. Intervals are 95% task-cluster bootstrap intervals.

Control	Rejections / trajectory	Task success (pp)
Random trigger ($A3$)	-0.031 $[-0.425, 0.356]$	-3.1 $[-6.9, 0.6]$
Observable trigger ($A4$)	$+0.013$ $[-0.200, 0.244]$	-1.3 $[-3.1, 0.0]$
Resampling only ($A6$)	$+0.231$ $[-0.119, 0.756]$	-1.9 $[-6.9, 3.1]$
Timing-matched random ($A7$)	$+0.194$ $[-0.063, 0.519]$	$+0.6$ $[-3.1, 4.4]$

sensitivity under two alternatives. The first assigns task failure, $Y = 0$, to an unsuccessful branch. Its rejection count remains the baseline count, because an incomplete branch has no defined full-trajectory count. The second drops a paired comparison if either arm failed on that baseline trajectory.

Recomputing the four direct $A2$ comparisons under all three conventions leaves every interval containing zero on both endpoints in both domains. For retail $A2 - A7$, the task-failure convention gives a success difference of 4.7 $[-0.3, 10.3]$ percentage points. Dropping its failed pair leaves 319 trajectories and gives 4.4 $[-0.6, 9.7]$ points. The main estimate is 4.4 $[-0.6, 9.4]$ points on 320 trajectories. The updated sensitivity includes $A7$, which the original six-arm supplement did not cover.

B.16 Trigger rates and the timing-matched arm

The random gate's firing probability is set to the observed attempted-trigger rate of $A2$ in each domain. This matches the rate in expectation, not exactly in a finite sample. Retail $A2$ attempts $208/320$ triggers (65.0%), compared with $194/320$ for $A3$. The airline counts are $93/160$ and $88/160$. Similar turn-level alert rates also need not give similar trajectory-level trigger rates. In retail, the observable gate attempts $294/320$ triggers (91.9%) at its frozen threshold.

The $A7$ design was written after the original six arms had been analyzed, but before $A7$ continuations were generated. It fixes the instruction, model, outcomes, and comparison with $A2$. Let M be the number of eligible generated assistant turns in a baseline trajectory. For an $A2$ trigger at zero-based index k, its relative position is $u = k/(M-1)$, with $u = 0$ when $M = 1$. The position pool contains the observed $A2$ values within that domain. For a triggered $A7$ trajectory, sample u from this pool and choose turn $\lfloor u(M-1) + 0.5 \rfloor$. This uses completed baseline trajectories, as described in Appendix A.3.

In the recorded runs, $A3$ and $A7$ attempt triggers on exactly the same trajectories. This set overlap was checked from the archived branch identifiers. Their relative-position medians are 0.589 and 0.000 in retail, and 0.525 and 0.000 in airline. The $A2$ median is 0.000 in both. Mean baseline turns remaining, including the trigger, are 16.5, 8.2, and 15.0 for retail $A2$, $A3$, and $A7$. The airline means are 12.5, 6.2, and 11.1. Remaining baseline turns are an exposure proxy, not the actual length of the new continuation. Matching relative positions reduces the average gap but does not equalize every trajectory's exposure. Figure 7 shows the trigger distributions and mean remaining baseline turns.

The additional $A3 - A7$ comparison holds the attempted-trigger trajectories and instruction fixed while changing the timing rule. Continuations still use arm-specific seeds. Its success differences are 5.3 $[0.6, 10.3]$ percentage points in retail and 3.8 $[1.3, 6.9]$ in airline. Its rejection differences are 0.028 $[-0.203, 0.253]$ and 0.225 $[-0.056, 0.563]$. These success intervals are the two that exclude zero among five direct contrasts, two endpoints, and two domains. They are exploratory and unadjusted for these multiple comparisons. They motivate a new timing study rather than identify a causal mechanism or establish that early triggering is harmful.

B.17 Gate quality on baseline turns

This post-hoc analysis reads frozen predictions on all $5{,}382$ retail and $1{,}929$ airline held-out baseline turns. Both predictors target r_3. The matched AUROC groups by task and generated assistant-turn index. At the frozen thresholds, the hidden gate alerts on 8.44% of retail turns and 9.90% of airline turns. The observable gate alerts on 8.68% and 8.76%.

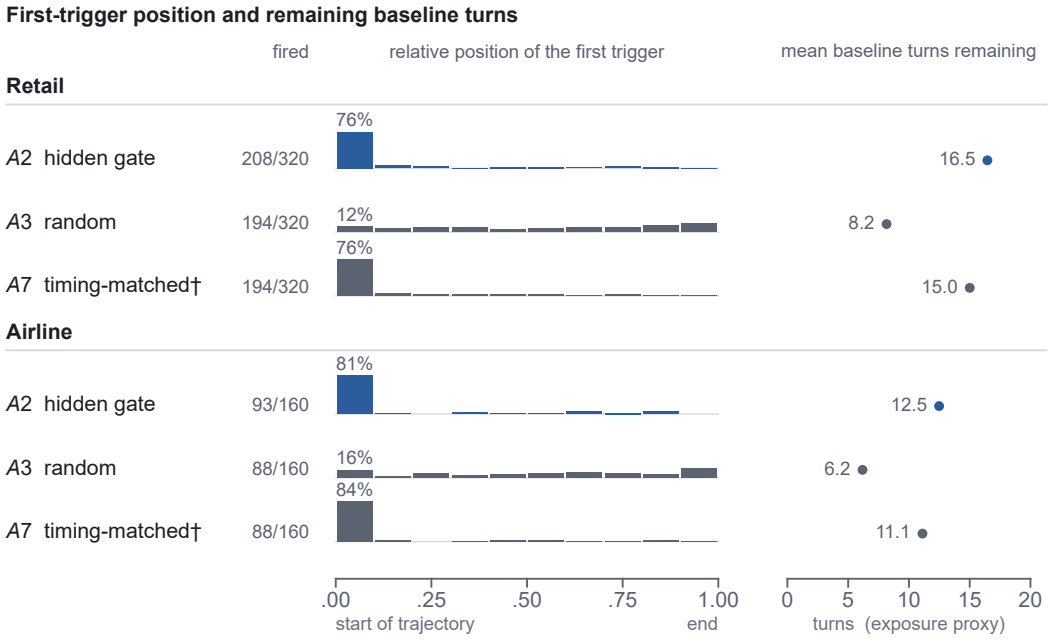

Figure 7: First-trigger timing on held-out baseline trajectories. Counts include all attempted triggers, including branches that failed to complete. Histograms show relative positions in ten equal-width bins, normalized within each arm's attempted triggers. Position runs from the first to the last eligible assistant turn. The dots give mean remaining baseline turns, including the trigger turn, over these same trajectories. These means are exposure proxies, not actual continuation lengths. $A7$ uses $A3$'s attempted-trigger trajectories and samples $A2$'s relative positions within each domain. It does not match exposure for each individual trajectory.

For an additional equal-alert comparison, we select the same number of highest-scoring turns from each predictor. That number is set by the hidden gate's realized alert rate. This changes the diagnostic operating point of the observable score, not the threshold used to run arm $A4$. The comparison does not regenerate any trajectories.

In retail, the pooled AUROC gain is 0.111 $[0.088, 0.135]$ and the task-and-turn-matched gain is 0.136 $[0.103, 0.170]$. Precision at the matched alert rate is 0.857 for hidden scores and 0.628 for observable scores. In airline, the pooled gain is 0.099 $[0.031, 0.163]$, while the matched gain is 0.108 $[-0.013, 0.191]$. Only 68 airline groups contribute to the matched statistic. The task-macro scores are 0.782 and 0.783, and ECE is slightly lower for the observable predictor. Thus, the evidence for a hidden-score advantage is not consistent across the airline views.

These measurements use every baseline turn, not just each trajectory's first threshold crossing. They also use outcomes under the original policy. They cannot determine the quality of decisions at actual first triggers or after verification has changed the trajectory. On training OOF data at a 10% alert rate, hidden precision and recall are 0.860 and 0.532 in retail. The corresponding observable values are 0.694 and 0.429 under fully nested evaluation. Appendix D explains the earlier optimistic training estimate for that comparator.

B.18 Completion-token sensitivity

Let N_a be the recorded full-trajectory agent completion-token count under arm a. For $Y_a \in \{0, 1\}$, the direct token-priced gain of $A2$ over $A3$ is

$$G_{2,3}(\lambda) = \mathbb{E}[Y_{A2} - Y_{A3}] - \frac{\lambda}{1000}\mathbb{E}[N_{A2} - N_{A3}]. \tag{14}$$

The price λ uses success probability on the $[0, 1]$ scale per $1{,}000$ completion tokens. It differs from the per-trigger cost κ in Eq. (8). This endpoint was added after the original runs.

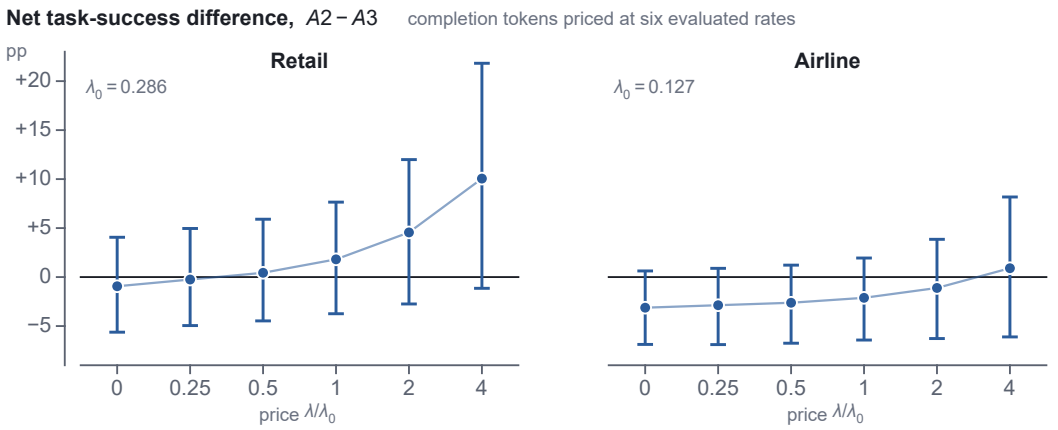

Figure 8: Post-hoc completion-token sensitivity for $A2 - A3$ on held-out 14B trajectories. Points show $G_{2,3}(\lambda)$ from Eq. (14) in percentage points; higher values favor $A2$. Bars are pointwise 95% paired task-bootstrap intervals. The six evaluated price ratios are spaced equally as separate cases. Connecting lines aid reading and do not form a confidence band. Each domain uses its own fixed reference price λ_0. The analysis excludes input tokens, latency, and hidden-state access costs, and does not compare $A7$.

Table 15: Post-hoc completion-token sensitivity for $A2 - A3$. Net differences are in percentage points of success, with pointwise 95% intervals. The scale λ_0 is defined separately within each domain. These intervals are not a simultaneous confidence band.

Price λ/λ_0	Retail	Airline
0	-0.9 $[-5.6, 4.1]$	-3.1 $[-6.9, 0.6]$
0.25	-0.3 $[-5.0, 5.0]$	-2.9 $[-6.9, 0.9]$
0.5	$+0.4$ $[-4.5, 5.9]$	-2.6 $[-6.8, 1.2]$
1	$+1.8$ $[-3.7, 7.6]$	-2.1 $[-6.4, 1.9]$
2	$+4.6$ $[-2.7, 12.0]$	-1.1 $[-6.3, 3.9]$
4	$+10.1$ $[-1.1, 21.8]$	$+0.9$ $[-6.1, 8.2]$

We define a reference price $\lambda_0 = 1000\,\overline{Y}_{A1}/\overline{N}_{A1}$ within each domain. It is 0.286 in retail and 0.127 in airline. This ratio sets a scale for the sensitivity analysis; it does not estimate the marginal value of a token. The six evaluated price multipliers are 0, 0.25, 0.5, 1, 2, and 4. The reference price is held fixed when forming the intervals.

The paired token difference is -96.2 per trajectory in retail and -79.3 in airline. Thus, increasing the price improves the point estimate for $A2$ relative to $A3$. The point estimates cross zero at $0.34\lambda_0$ and $3.10\lambda_0$. This arithmetic does not show that earlier triggering causes token savings. Every pointwise interval at the six evaluated prices contains zero (Table 15 and Figure 8). We make no simultaneous claim across a continuous price range.

For this write-up, we recompute the curve using the same 2,000 task-cluster draws as the primary $A2 - A3$ success contrast. Each draw retains paired success and token differences and is reused at every price. The zero-price interval therefore matches Tables 5 and 14 exactly. The point estimates and conclusions agree with the earlier cost archive. Small interval changes reflect the synchronized bootstrap draws. Neither this comparison nor the earlier cost sweep includes $A7$.

Completion tokens measure only one resource. They exclude input tokens, wall-clock latency, and the cost of storing or exposing internal states. The pre-action probe needs no additional generated text, but its extraction, storage, readout, and serving costs were not measured. The replay-and-branch study is therefore not a full deployment-cost evaluation.

C AN ATTEMPTED CROSS-FAMILY REPLICATION

We also tried to test whether the pre-action comparison holds in a second model family. We wrote the design before running it. The layer, the regularization strength, the ranking measure, the comparison predictors, and the task and call-position groups were all fixed in advance. Each model would receive its own probe, fitted on its own corpus. A candidate would be accepted on corpus quality alone and never on a probe result. We also fixed the order of candidates in advance and would use the first model that passed.

Three candidates failed, each for a different reason. We report them because they show what makes this comparison hard.

The first candidate was an 8B model from a second family. It had 70% of its tool calls rejected. Only 61% of its argument values appeared anywhere in the earlier conversation, compared with 89% to 92% in our corpora. The model wrote placeholder identifiers instead of reading them from the context. Its rejection label therefore describes a different event from ours.

The second candidate was a 24B model from a third family. It matched our corpora closely in pass rate, rejection rate, and argument grounding. However, it issued several tool calls in one turn on 19% of its calling turns. Our groups fix the task and the exact call position. In its corpus, one calling turn could hold up to five calls, while a calling turn in our corpora almost always holds one. We did not relax the acceptance criterion after seeing this.

The third candidate passed every corpus criterion but could not run the fixed configuration. It has 24 transformer layers, so our fixed layer index of 25 does not exist in it. Choosing a new layer would turn the replication into a new round of tuning. The design was written to prevent exactly this. The model also writes reasoning tokens before its action. A null result would then be ambiguous. It could reflect a family effect, or a model that moves the relevant computation past our observation point.

We therefore report no cross-family result. Two lessons may help future attempts. First, check that the fixed configuration can run on a candidate before checking corpus quality, because a layer index does not transfer across model depths. Second, a small pilot cannot screen a corpus for a measure that needs groups the pilot does not contain. Our 20-task pilot had no group with both labels, so it could not test the quantity we cared about.

D ANALYSIS AUDIT

We record corrections that affect reported numbers.

First, one bootstrap routine built a cluster key by joining a task identifier and a replicate index as text without a separator, so two different clusters could collide. We replaced it with an arithmetic encoding. The collision fired in about 5% of replicates and merged one cluster each time. Recomputed intervals match the earlier ones to three decimals, and no conclusion changes.

Second, the held-out models for r_1 and r_2 were fitted after the intervention stage, with all settings fixed beforehand and no tuning on held-out data. The r_3 gate used for the intervention was frozen before the held-out runs. We keep this distinction rather than presenting all three as equally prospective.

Third, an earlier version of the observable gate report fitted text vocabularies outside the cross-validation loop and evaluated its stacker on the same scores used to fit it. We repeated that audit with fully nested folds. This lowered its training AUROC from 0.837 to 0.834 in retail and from 0.810 to 0.762 in airline. The frozen artifact itself was not changed, since it had already been applied to held-out data. The same leak affected the text comparison predictors used for the paired gains. Their vocabularies are now fitted inside each training fold, and every gain and interval we report comes from that rerun. The rerun changed the hidden-state gain over the stack from $+0.1151$ to $+0.1146$ in retail and from $+0.0845$ to $+0.0909$ in airline. The gains after the call is written changed from $+0.0080$ to $+0.0081$ and from -0.0099 to -0.0054. No conclusion changes.

Fourth, an exploratory regression of the change in success on the change in writes and rejections is reported only as a diagnostic. Both regressors are measured after the intervention, so conditioning

on them can introduce bias. With arm and domain effects included, only the rejection coefficient stays away from zero.

Fifth, the cost-weighted endpoint in Appendix B.18 was not fixed in advance. The frozen protocol names rejections per trajectory and task success. We added the price sweep after those runs, so we mark it exploratory and read it only as a sensitivity check on the pre-registered null. Every price in the sweep uses one fixed set of bootstrap resamples, so the reported curve is a single coherent object rather than one draw per price.

Sixth, the intervention intervals were first computed with a single random generator shared by every bootstrap in one run. Each interval then depended on what else the run had computed before it. Adding the timing-matched arm shifted the endpoints of earlier intervals by about 0.02, for example from $[-0.459, 0.125]$ to $[-0.441, 0.100]$ for $A2 - A3$ in retail. No point estimate changed, since point estimates use no random draws, and no conclusion changed. We now seed each bootstrap from its own inputs, so an interval no longer depends on its neighbors, and every intervention interval in this paper comes from that version. We checked that adding the new arm left the set of paired trajectories and the three earlier retail contrasts unchanged.

D.1 Paired target comparison: implementation and reporting

A writing-stage audit found two departures from the planned target comparison. The protocol requested trajectory-weighted scaling. The implementation fits unweighted means and variances within each training fold. It uses trajectory weights for classifier fitting and inner log-loss selection. The same scaling procedure applies to both targets, but its effects need not be equal. A weighted-scaling sensitivity analysis has not been run. The planned Brier and log-loss evaluation against training-rate constant predictors was also not completed in the archive. Inner log loss was used for tuning, which is a separate operation. We therefore report ranking results for this additional analysis and do not claim a probability-quality comparison. The archived fits and scores were retained.

E Discussion and limitations

E.1 Discussion

The value of an internal signal depends on when it is read and what it predicts. Before generation, the hidden probe improves rejection ranking over the tested text baseline in all four model and domain settings. After the call is written, neither 14B comparison shows a detectable gain over the observable predictor. This supports pre-action monitoring in the settings studied. It does not establish a universal advantage over text-based methods or equivalence after generation.

The paired target comparison finds a ranking gap within the tested pre-action layer. In retail, its gain over the text baseline is also larger for rejection than for final failure. This empirical result does not establish an absence of task information. The theory concerns the narrower input of binary execution records. Binary calling and rejection labels omit action content and task requirements. A shared predictor can therefore fail when those omitted factors change the meaning of a record. Our transfer bound states the required distributional condition. It does not place an upper bound on task-success prediction from full hidden states or richer score sequences. The terminal probe for reference-write completion provides evidence that a richer representation retains some task-related information.

The control study adds a separate requirement: the chosen intervention must help where it is applied. Better baseline rejection ranking in retail does not establish such a benefit. The direct comparisons find no detectable advantage for the tested verification policy, including against the timing-matched control. That control reduces the exposure gap without matching every trajectory. The completion-token analysis concerns the original random control and six price points. Together, these results motivate direct evaluation of a monitoring policy's outcomes, beyond its predictive performance.

E.2 LIMITATIONS

The evidence covers two domains of one benchmark and two sizes of one model family. The intervention study uses only the 14B model. The attempted cross-family replication was not completed (Appendix C). Transfer to other model families and live systems remains untested.

The text baselines omit policy text and tool definitions that are constant within a domain. Richer predictors may benefit from interactions with these inputs.

Only groups containing both accepted and rejected calls contribute to matched AUROC. Many groups are small. Retail contributes 303 groups from 39 of 40 test tasks; airline contributes 55 groups from 12 of 20. The airline estimate therefore covers fewer tasks and has a wider interval. The test AUROCs exceed the OOF values, but the task sets and fitted probes differ. Their order does not show that prediction becomes easier on unseen tasks. The bootstrap intervals hold fitted scores fixed and exclude uncertainty from model selection and refitting.

The paired target comparison uses one layer originally selected for rejection prediction. Both hidden targets choose the strongest regularization in the tested grid. The shared groups cover 47.5% of retail calling prefixes and 13.2% of airline calling prefixes. Only 12 airline tasks contribute, so those results remain descriptive. The analysis does not test symmetric layer selection or provide a new independent test-set confirmation. Future interactions also affect final outcomes, and the observed gap does not isolate their contribution.

Repeated-prefix sampling uses one temperature and 16 responses per prefix. The all-turn panel covers retail only, and the rejection audit executes only the first parsed call. Target-specific eligibility also changes the effective sample. The panels do not separate state uncertainty from context-based prediction error.

The control results concern one instruction and one training alert level. The replay-based timing controls use completed baseline trajectories and do not specify ready-to-use online policies. Airline has limited power for the rejection endpoint. Input-token cost, internal-state access, and serving overhead were not measured. Several diagnostics were added after the original runs and are marked as such. Confidence intervals that include zero do not establish equivalence.