GUARDVALUE: ESTIMATING ONE-STEP VALUES AND POLICY EFFECTS WITH CLONED TRAJECTORY PREFIXES

Anonymous authors
Paper under double-blind review

ABSTRACT

Pre-execution guards decide whether a tool-using agent's proposed action runs. Comparing guarded with unguarded episodes measures a policy that may intervene repeatedly, not the value of one intervention. We introduce GuardValue, which clones a live benchmark episode to measure both from a shared history and sums one-step values over guarded-policy visits, as the performance-difference identity prescribes. In a ten-arm matched-budget audit on 278 τ^2-bench tasks, type-blocking raises success by 0.264 where the reference uses no irreversible action and lowers it by 0.101 elsewhere. One silent refusal has near-zero value, while repeated refusal changes success by -0.089 per first mutating proposal in retail, $+0.270$ in airline and $+0.227$ with a second actor. With every visit measured under deterministic serving and synchronized starts, the occupancy sum equals separately measured effects task by task in four checks covering silent and noticed refusal and a budget-charged engine, and on 47 of 50 tasks without it. A fixed visit schedule leaves the registered silent test unresolved, and production-serving airline attempts disagree. For the second actor, first-point values rank the engine above silent refusal, whose effect is ten times larger. First-point values should not be reported as policy effects.

1 INTRODUCTION

Tool-using language-model agents modify persistent state in databases, repositories and web applications (Yao et al., 2023; Schick et al., 2023; Barres et al., 2025; Zhou et al., 2024; Jimenez et al., 2024; Trivedi et al., 2024). A mistaken write can remain harmful after the model detects it. Pre-execution guards screen proposed calls (Cuadron et al., 2025; Yuan et al., 2026; Rovai, 2026), while contracts and specifications constrain tool use (Babu & Iyer, 2026; Wang et al., 2026; LeVine et al., 2026). Prior evaluations compare guarded with unguarded agents on the same tasks (Cuadron et al., 2025; Sah et al., 2026; Li, 2026). Charged checks consume slots that could support replanning, confirmation or candidate selection (Shinn et al., 2023; Huang et al., 2025). We therefore first compare ten guard and comparison arms under matched interaction budgets on 278 τ^2-bench tasks. Relative to NoGuard, blanket type-blocking changes task success by $+0.264$ when the reference trajectory uses no irreversible action and by -0.101 when it does (Figure 1(a), Table 1).

However, a comparison of separately run episodes measures a guard policy that may intervene at every mutating proposal, not the value of any single intervention. Once the guarded and base trajectories diverge at the first block, later outcomes reflect every subsequent decision (Ro et al., 2025; Pyskło et al., 2026). A proposal-level decision, including whether to review it, requires the local value and review cost. Failure-attribution methods re-run parts of recorded trajectories to localize an error (Shah, 2026; Zhang et al., 2026; Bonagiri et al., 2026), but a local value need not equal the effect of a policy that keeps intervening. The established performance-difference identity expresses that effect as a sum of one-step values over states the guarded policy visits (Kakade & Langford, 2002; Schulman et al., 2015), and a measurement at the first intervention omits every later term.

To measure both quantities on one history, we introduce GuardValue, an instrument that clones a live, stateful benchmark episode at a proposed mutation, including its threaded execution and budget state. Branching from a shared state is established in TRPO's vine procedure (Schulman et al.,

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

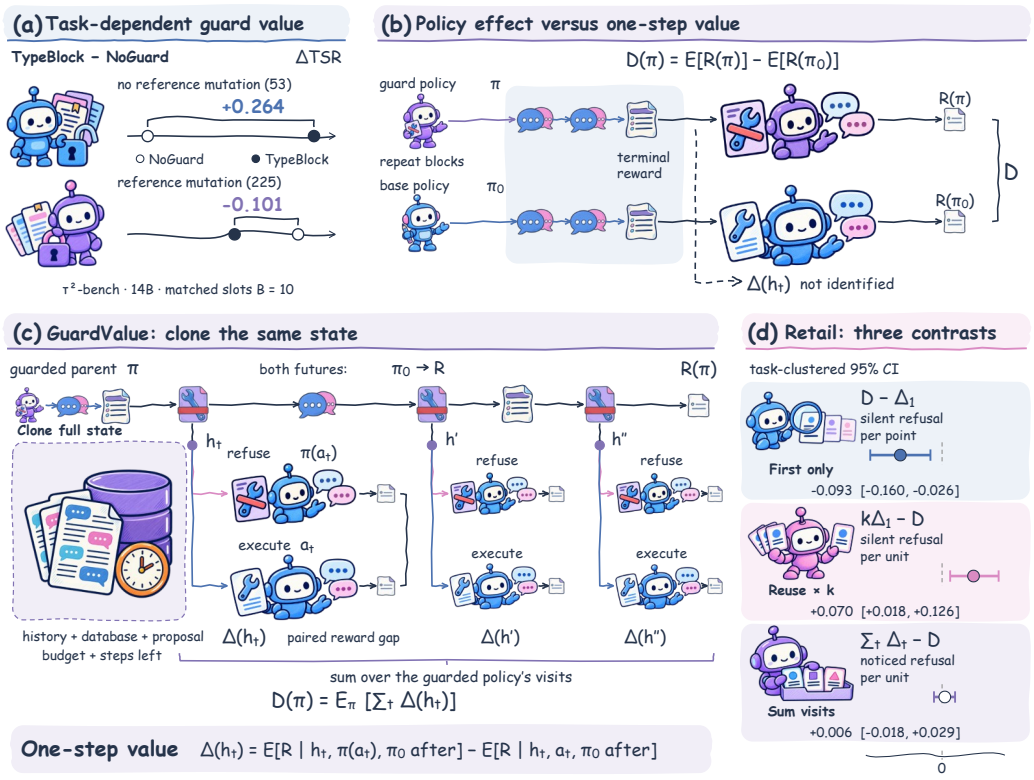

Figure 1: **GuardValue separates a policy effect from one-step intervention values.** **(a)** At matched budget, TypeBlock changes task success relative to NoGuard in opposite directions across reference-mutation strata. Positions are schematic. **(b)** Separate guarded and base episodes estimate the whole-episode policy effect $D(\pi)$ after their trajectories diverge. **(c)** Cloning the decision state isolates one-step values, and repeating the comparison at guarded-policy visits estimates their sum. **(d)** On retail, silent refusal fails the registered first-point test and a post hoc firing-count analysis on occupancy walks, while the noticed-refusal sum is compatible at the measured precision. Error bars are task-clustered 95% bootstrap intervals, and filled markers exclude zero.

2015). One clone intervenes and another executes, after which both follow the base policy, so their paired terminal rewards estimate a one-step value. A policy arm started from the same prefix keeps intervening. Measuring the one-step value at every guarded-policy visit estimates the performance-difference sum. The identity holds exactly in expectation. A gap between the measured sum and the measured policy effect therefore reflects clone fidelity, unmeasured points, variation between runs or sampling error, rather than a failure of the identity. We check clone fidelity separately with deterministic fixtures, paired execute controls and a serving-determinism diagnostic.

Across 270 retail and 115 airline branch points from 342 and 150 task-seed units, one silent refusal has near-zero value, while refusing throughout changes success by -0.089 in retail and $+0.270$ in airline per first mutating proposal. A second actor replicates the airline gap. With every visit measured under deterministic serving and synchronized episode starts, occupancy sums equal separately measured effects on every task for silent refusal on 25 informative retail tasks and for noticed refusal and the reviewed engine on airline. With the second actor, first-point and count-scaled values rank the engine above silent refusal, whose whole-episode effect is ten times larger.

Our contributions are as follows:

1. We introduce GuardValue, a within-prefix instrument that clones a live benchmark episode so that one-step values and policy effects share a history. Under deterministic serving,

clones reproduce recorded continuation paths in 428 of 430 comparisons and their rewards in all of them, and its policy-effect estimates show no detected difference from between-episode estimates (Sections 5 and 6).

2. We reject silent refusal's first-point substitution in pre-registered tests in two domains, whose gaps have opposite signs, and for a second actor. A post hoc analysis also rejects firing-count scaling for silent refusal on retail, and with the second actor both substitutes misorder silent refusal and the reviewed engine (Section 6).
3. We test instrument fidelity and coverage where the identity predicts equality. With every visit measured, deterministic serving and synchronized episode starts, the occupancy sum equals the separately measured effect on every task in four checks covering silent refusal, noticed refusal and a budget-charged reviewed engine with two actors, while a fixed visit schedule and production-serving campaigns leave it unresolved (Sections 6.3 and 6.4).

2 Related work

Pre-execution guards and the cost of blocking. Shields use temporal-logic specifications to correct unsafe actions (Alshiekh et al., 2018; Könighofer et al., 2023), and safe reinforcement learning measures the resulting performance trade-off (Jansen et al., 2020; Banerjee et al., 2024; Dawood et al., 2025). For tool agents, SABER-style guards block or route state-mutating calls (Cuadron et al., 2025), AEGIS adds a pre-execution firewall (Yuan et al., 2026), and specifications constrain calls before they run (Babu & Iyer, 2026; Wang et al., 2026; LeVine et al., 2026). CIVeX executes a state-changing action only when its causal effect is identifiable, certified by a lower confidence bound and within provenance and risk limits (Rovai, 2026). It decides whether to act, while GuardValue measures what a fixed intervention changes at a particular state of a cloned live episode.

Verification and intervention budgets. A safety mediator can intercept up to 94% of non-compliant actions while increasing token use by 2.0 to 2.8 times (Sah et al., 2026). Test-time compute can be allocated adaptively (Snell et al., 2025), including best-of-N selection (Huang et al., 2025). Verification budgets have been allocated per problem (Dip et al., 2026), across workflow nodes (Ro et al., 2025), over time (Patel, 2026), and by the myopic counterfactual harm of one verification opportunity per episode (Tang & Zhan, 2026). In that setting, the one-step value is the policy effect. That equality need not hold once a guard can intervene repeatedly. Under a medium safety budget, a budget-aware planner reduces unsafe completion from ReAct's 20.8% to zero but completes 28.6% of tasks against ReAct's 76.2% (Li, 2026).

Attributing a failure inside a trajectory. Sherlock injects faults at workflow nodes (Ro et al., 2025). Causal Agent Replay applies interventions to recorded runs (Shah, 2026), AgenTracer replaces actions with oracle guidance (Zhang et al., 2026), CausalFlow repairs a step and re-executes (Bonagiri et al., 2026), and Counterfactual Trace Auditing aligns separately run traces (Zhou et al., 2026). Self-correction similarly revises a step after criticism (Shinn et al., 2023; Madaan et al., 2023; Gou et al., 2024). GuardValue clones the live environment at the decision state, applies a fixed intervention in one branch, follows both branches with the same base policy, and repeats the measurement along the guarded policy's own visits. The performance-difference identity and shared-state branching, as in TRPO's vine procedure, are established (Kakade & Langford, 2002; Schulman et al., 2015). Our contribution is the live-agent instrument with threaded execution and budget state, its measured fidelity and failure modes, and the finding that first-point values misstate and misorder repeated guard policies. We use verified τ^2-bench (Yao et al., 2025; Barres et al., 2025; Cuadron et al., 2025). Its reference trajectory identifies action use rather than necessity. Separate-arm evaluation is the appropriate design for task-pool policy performance, and Section 4 shows why it cannot identify a local effect.

3 A matched-budget audit

We compare guard and baseline arms under the same per-episode cap B on interaction slots (Algorithm B.1 in Appendix B). A Qwen2.5-14B-Instruct actor (Qwen et al., 2025) runs all airline (50), retail (114) and telecom (114) tasks of verified τ^2-bench for three seeds at $B = 10$, producing 8,340 completed episodes over ten arms (Appendix O).

Table 1: Task success by reference-mutation stratum at matched budget $B = 10$, three seeds, over the complete pool of 278 tasks. A task belongs to the reference-mutation stratum when its reference trajectory executes an irreversible action. The label records use, not necessity. V is TypeBlock minus NoGuard, with task-clustered 95% intervals $+0.264\,[+0.164, +0.371]$ and $-0.101\,[-0.133, -0.070]$. StrictWr is StrictWrite and NecGate is NecessityGate. Bold marks the best non-oracle arm in each row.

Stratum	Tasks	NoGuard	TypeBlock	StrictWr	NecGate	Oracle
No mutation	53	0.277	**0.541**	**0.541**	0.352	0.541
Mutation	225	**0.127**	0.027	0.081	0.121	0.154
All	278	0.156	0.125	**0.169**	0.165	0.228

Arms and budget. Ten arms cover NoGuard, blanket type-blocking, zero-overhead and strict-write variants, NecessityGate, reflection after a block, replanning, confirmation, best-of-three selection and a non-deployable, label-informed oracle (Appendix C). TYPEBLOCK blocks every action in the benchmark's irreversible-action taxonomy, and STRICTWRITE removes the escalation operation from that taxonomy. NECESSITYGATE spends one language-model decision on each irreversible proposal (Appendix D), and REFLECTREPAIR also tells the actor why a call was blocked (Appendix E). A check, confirmation, replan, candidate rollout, verifier call or executed tool costs one slot. Allocation is episode-local. Episodes are paired across arms, and proposals beyond the cap fail at the pre-proposal state. Matched slots do not equalize token use, which averages 56.1k to 66.7k per episode (Table G.1 in Appendix G).

Metrics and strata. Task success rate (TSR) and mutation-associated failure rate (MAF) are episode means. MAF counts failed episodes in which a taxonomy mutation executed and does not establish causation. The guard value of arm a is its matched-budget success gap $\widehat{V}_a = \widehat{\mathrm{TSR}}_a - \widehat{\mathrm{TSR}}_{\mathrm{NoGuard}}$. A task belongs to the reference-mutation stratum when its reference trajectory executes an irreversible action. This label records use rather than necessity and is unavailable to a deployed guard. The two stratum contrasts were pre-registered as confirmatory, and every other audit comparison is exploratory. Intervals are task-clustered bootstrap intervals that keep the three seeds together. The expanded audit and the capability labels are defined in Appendices I and L.

The guard-value sign flip. TypeBlock raises success from 0.277 to 0.541 on the 53 tasks whose reference uses no irreversible action and lowers it from 0.127 to 0.027 on the other 225 (Table 1). Both confirmatory contrasts exclude zero, and the flip persists with a 7B actor, $+0.214$ versus -0.095 (Appendix M), and under taxonomy perturbations (Appendix K). Pooled over the mixture, the effect is unresolved, with a task-micro gap of $-0.031\,[-0.068, +0.006]$ and an equal-domain gap of $+0.022\,[-0.022, +0.066]$. The domain effects, $+0.253$ on airline, -0.117 on retail and -0.070 on telecom, reflect each domain's mix of strata and its success rates (Appendices F and H).

Deployment implications are equally conditional. TypeBlock reduces MAF from 0.644 to zero by construction, with a break-even risk weight of 0.048 against NoGuard (Appendix G). A cross-fitted choice between TypeBlock and NecessityGate without task features selects TypeBlock in every fold (Appendix J). On 11 capable AgentDojo (Debenedetti et al., 2024) mutation tasks, a test of the mutation stratum only, TypeBlock lowers success from 0.73 to zero (Appendix N). Appendix A collects these results. The audit therefore finds opposite average effects in retrospectively labeled strata that deployed guards cannot observe. It cannot identify the value of one blocked action.

4 TWO ESTIMANDS

Scope of separate-episode comparisons. Section 3 compares separately run episodes, which is appropriate for policy performance on a task pool (Sah et al., 2026; Cuadron et al., 2025; Li, 2026). It cannot identify the effect of blocking one action. At step t the agent proposes an action a_t. We write h_t for the full decision state after that proposal, namely the conversation history, database, pending proposal, spent budget and remaining horizon. The guard policy π maps h_t to allow, block or replace. The unguarded base policy π_0 always allows, and R is the terminal reward.

The one-step value. The value of intervening at h_t, and only there, is

$$\Delta(h_t) \;=\; \mathbb{E}[R \mid h_t, \pi(h_t), \pi_0 \text{ after}] \;-\; \mathbb{E}[R \mid h_t, a_t, \pi_0 \text{ after}]\,, \tag{1}$$

where both continuations run π_0 to the end of the episode. A notice makes a refusal part of the agent's observed history. Without it, a silent refusal may only delay execution. The recorded continuations do not identify the next proposed action, so we cannot count repeated calls.

The policy effect. Section 3 reports a different quantity,

$$D(\pi) \;=\; \mathbb{E}[R(\pi)] - \mathbb{E}[R(\pi_0)]\,, \tag{2}$$

an expectation over whole episodes in which π intervenes wherever it fires. The audit estimates it for charged guard checks over task pools, and the branch experiments estimate it for refusal policies and a charged reviewed engine over task-seed units that share a prefix.

The connecting identity and two substitutions. Suppose both policies start from the same task and seed, share the environment, simulated user, reward and horizon, and differ only in the guard's response to mutating proposals. The finite-horizon performance-difference identity (Kakade & Langford, 2002; Schulman et al., 2015) then gives $D(\pi) = \mathbb{E}_\pi\big[\sum_t \Delta(h_t)\big]$, where the expectation follows the guarded policy's own trajectories and the guard's response includes any slot it charges. When the response, including its charge, matches π_0's, $\Delta(h_t) = 0$. An uncharged refusal policy that allows meets this condition. A reviewed allow consumes a slot and need not have zero value, so engine walks measure every reviewed proposal. The identity holds exactly under these assumptions. With every visited point measured on a faithful clone, the occupancy residual $R_{\text{occ}} = \sum_t \Delta_t - D$ has expectation zero. A nonzero measured residual therefore reflects clone fidelity, coverage of visited points, matched support between the sum and D, variation between campaigns or sampling error.

For task-seed unit i, let $\Delta_{1,i}$ be the one-step value at its first mutating proposal and k_i its episode's firing count for π (overrides for the engine). Both are zero when no mutating proposal occurs. Two approximations replace the sum with the first point alone,

$$D(\pi) \;\approx\; \overline{\Delta}_{\text{first}} = \mathbb{E}_i\big[\Delta_{1,i}\big], \tag{3}$$

or with the first point scaled by the unit's firing count,

$$D(\pi) \;\approx\; \mathbb{E}_i\big[k_i\,\Delta_{1,i}\big]. \tag{4}$$

The first ignores later interventions. The second assumes that every later intervention has the first one's value. A first intervention may only delay an action that the base policy proposes again, while repeated intervention prevents it. Scaling by k_i can also fail when later decision states differ in remaining horizon, conversation history or action value. Section 6 tests both substitutions, comparing the first with a policy effect measured on the same branch points, and Section 6.3 estimates the occupancy sum and reports its coverage. Appendix R.1 maps every estimator to its unit-level form, records and support.

5 GUARDVALUE: A WITHIN-PREFIX INSTRUMENT

GuardValue estimates Equation (1) from two continuations of one prefix, which this benchmark does not natively offer. The benchmark is stateful, offers no rollback, and runs its conversation with a simulated user on a background thread.

Branching a live episode. At a decision point we deep-copy the environment and replace the running episode's synchronization primitives with fresh ones. The copy resumes where it was waiting for an action rather than at the start of the step (Figure 1(c) and Appendix P). Each branch has its own complete environment, including database state, conversation, simulated-user state and evaluator. The parent episode continues undisturbed. Each continuation receives the pending proposal and remaining horizon, with its own copy of the parent's budget record, and calls the actor and customer again without copying any server-side generation state. An execute branch submits the pending proposal. An intervention branch blocks or replaces it. Both then resume in the episode loop used for whole episodes. A second execute branch measures re-run variability without changing the estimand.

Matched budget and one scorer. Branches share the per-episode interaction cap of Section 3. A refusal executes nothing and charges no slot, while the reviewed engine's review is a model call charged against the same budget. The harness also has a separate intervention budget, uncapped in every experiment here (Appendix Q). Branches and whole episodes are scored by one function. A one-step arm continues under the base policy, so it measures Δ. A policy arm applies the guard to every later mutating proposal as well, so repeated prefixes yield paired estimates of both quantities.

Clone conformance. Table Z.1 in Appendix Z lists the checks on the clone path. Fixtures reproduce an uninterrupted run and live execute pairs agree in reward, but 4 of 273 adjacent comparisons in the airline technical repeat disagree. A registered diagnostic separates serving variation from an error in the copy (Appendix Z.2). On the production server, which caches prompt prefixes and batches requests, 14 of 444 re-sent requests differed from their in-run responses. Two re-sends agreed with each other on either server, consistent with output that depends on the server's in-run state. With prefix caching off and one sequence per batch, 428 of 430 airline clone comparisons over three campaigns followed identical recorded paths, and all 430 reached the same reward. The two exceptions began from identical states when the benchmark's threaded loop returned an observation in a different form. The same loop occasionally returns from reset before the customer's opening message, in 6 of 212 runs, which a clone never passes through. The remaining variation is consistent with thread timing and serving state rather than a copy error.

6 BRANCH EXPERIMENT RESULTS

Each study's decision rule was registered before its decisive run, and Appendix R records amendments, deviations and the later technical repeat with what was known at the time. An early implementation ended branches at the agent's first message to the simulated customer, before the end of the episode. Those estimates are excluded from every claim here and retained with the defect identified (Appendices S and Y). All results below use the corrected full-horizon loop. Figure 2 shows the three comparisons for each domain, actor and guard.

Design. The retail study clones 342 task-seed units at the first mutating proposal (Appendix U). One-step arms intervene once and then follow π_0, while policy arms keep intervening, and Δ_1 denotes the one-step value at that first proposal. All units complete. Among 270 branch points, 51 are informative (some branch scores above zero), exceeding the pre-registered floor of 25. The airline study repeats the design on 150 units, 115 of which reach a branch point (Appendix V).

Within-prefix and between-episode policy effects do not differ detectably. Refusing every mutating proposal changes retail success by $-0.070\,[-0.126, -0.018]$ within prefixes and $-0.062\,[-0.118, -0.009]$ between episodes. Their difference is $-0.008\,[-0.035, +0.020]$ (Appendices T and U). No equivalence margin was specified, so this shows no detected difference, not equality.

6.1 REFUSING ONCE AND REFUSING ALWAYS, FROM THE SAME PREFIX

On retail, one silent refusal is near zero, $+0.004\,[0.000, +0.012]$, while refusing throughout costs 0.089 per branch point. Their pre-registered gap is $-0.093\,[-0.160, -0.026]$. The count-scaled approximation also misses the policy effect in the post hoc walk analysis, $+0.070\,[+0.018, +0.126]$. Figure 2(b) forms the firing-count product from branch records, while the walk form multiplies the first value by the visit count K of the separate occupancy walks, whose first states and counts differ (Appendix R.1). A noticed refusal changes the history and has a smaller gap, $-0.037\,[-0.085, +0.004]$. This interval includes zero under a width rule fixed in advance (Appendix R), but the rule is not an equivalence test.

The silent one-step arm changes one of 270 branch outcomes, while the repeated policy changes 38, including 31 losses. At 229 points refusing adds exactly one step, consistent with delay (Appendix U). With notice the policy intervenes 2.53 times per branch point instead of 23.18 (Figure 2(e)), so the notice changes both the trajectory and the support on which later interventions are measured.

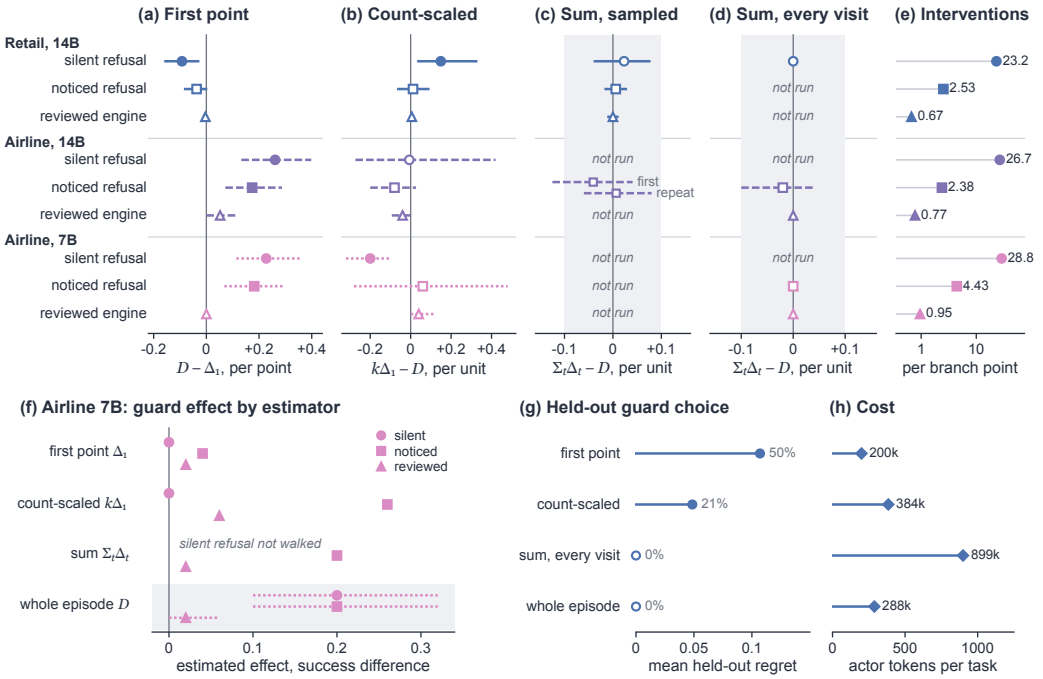

Figure 2: **First-point values can misstate a repeated guard policy, and sums over every visit agree with its effect.** Rows show the 14B actor on retail and airline and the 7B actor on airline. **(a)** Policy effect minus first-point value per branch point. **(b)** First-point value times the unit's firing count, minus the policy effect, per task-seed unit. **(c)** Occupancy sum minus policy effect on the registered sampled walks, with the first airline attempt and its technical repeat. **(d)** The same residual with every visit measured under deterministic serving. **(e)** Mean interventions per branch point, overrides for the engine. **(f)** Second actor, each guard's mean effect over 50 tasks under four estimators, with intervals for the whole-episode effect. **(g)** Post hoc held-out regret, the best guard's held-out effect minus the chosen guard's, over 1,000 random half-task splits. Shares indicate how often the 2,000 choices select a worse guard. **(h)** Actor tokens per task consumed by each estimator's inputs. Intervals are task-clustered 95% bootstrap intervals, filled markers exclude zero, gray bands mark the registered ± 0.10 tolerance, and Table R.1 lists the values.

6.2 A SECOND DOMAIN, WHERE REFUSING HELPS

On airline, refusing throughout helps by $+0.270$ per branch point, while one silent refusal is $+0.009$. Their pre-registered gap is $+0.261\,[+0.133, +0.398]$. The within-prefix and between-episode policy estimates differ by $+0.007\,[-0.047, +0.067]$. The firing-count approximation remains unresolved on airline, with $-0.007\,[-0.273, +0.420]$ per unit. Silent first-point substitution is therefore rejected in both domains, in opposite directions, and these data do not isolate why.

The noticed airline refusal has value $+0.096$ against the policy's $+0.270$, a gap of $+0.174\,[+0.072, +0.287]$ from a comparison not designated to decide the hypothesis. A pre-registered pooled comparison combines it with retail's negative gap and includes zero through cancellation, so we report that test as mis-specified (Appendix Y). The reviewed engine completes 1.85 reviews and 0.67 overrides per branch point in retail, and 2.23 and 0.77 in airline. Its retail point estimates lie within 0.012 of zero because it rarely changes an action, not because review is harmless.

6.3 SUMMING OVER THE POINTS THE GUARDED POLICY VISITS

The occupancy walks follow each guarded policy and clone at its visits (Appendix W, supports in Table W.1). Every-visit agreement tests instrument fidelity and coverage, not the identity that

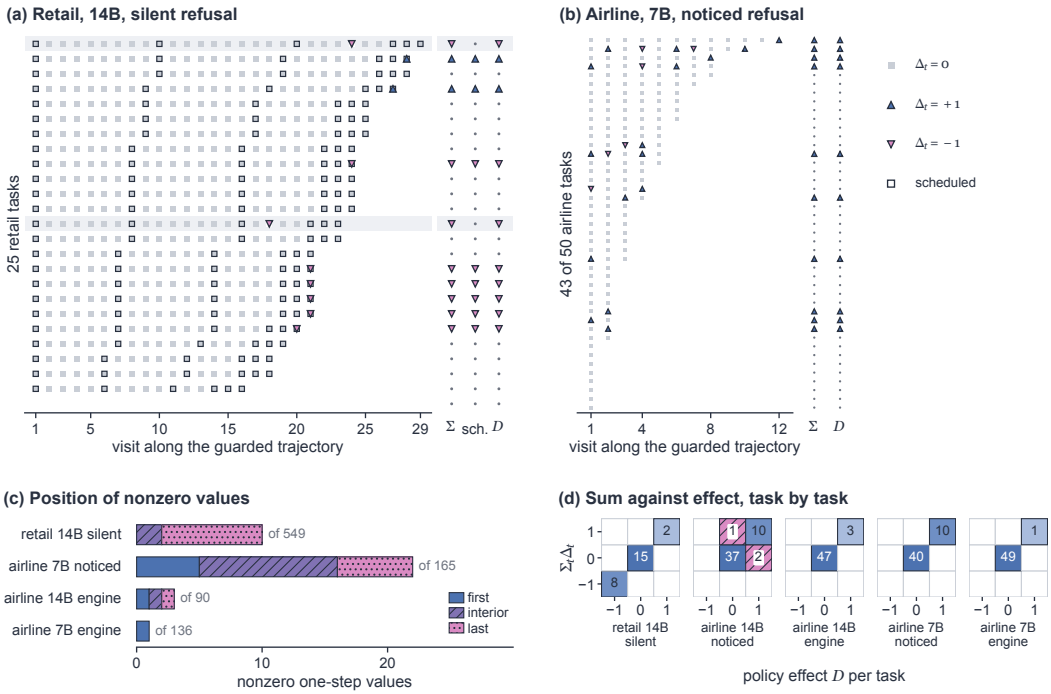

Figure 3: **Every-visit walks locate the policy effect after the first intervention, and their sums equal it task by task apart from three empty-opening tasks.** **(a)** One-step value Δ_t at every silent-refusal visit on 25 retail tasks under deterministic serving, ordered by visit count. Outlined cells mark the registered four-point schedule and its terminal margin. Right columns show full sums, scheduled estimates and policy effects, shading the two tasks with interior effects missed by the schedule. **(b)** Noticed refusal with the 7B actor on the 43 of 50 airline tasks with a visit. **(c)** Nonzero one-step value positions among all measured visits in four every-visit studies. **(d)** Taskwise occupancy sums versus policy effects in the five every-visit checks, with agreement on the diagonal. The three off-diagonal airline tasks started from an empty opening message (Appendix Z.3).

predicts it. The registered primary silent-refusal schedule measures four positions placed by a preliminary visit-counting pass, plus the last three counted visits. Its occupancy residual remains unresolved, $R_{\mathrm{occ}} = +0.023\,[-0.040, +0.078]$ on 325 units with width 0.118, while on those units the count-scaled first value misses the policy effect, $+0.065\,[+0.012, +0.120]$. Seventeen units lose their last point with outcome-dependent missingness (Appendix W). A later registered study measured every silent visit under deterministic serving on the 25 retail tasks with informative first branch points. The occupancy sum then equaled the measured effect, $-0.24\,[-0.48, 0.00]$, on every task (Figure 3(a)). Descriptively, 10 of its 549 visits had a nonzero value, none at the first visit, eight at the last and two in the interior (Figure 3(c)). On the same walks, the registered schedule misses two unscheduled interior effects and gives $+0.08\,[0.00, +0.20]$, which does not resolve. On this subset the unresolved silent result reflects the schedule rather than the instrument, although full coverage takes 3.8 times as many clones (Appendix Z.5).

Measured at every visit, the noticed refusal's occupancy sum is compatible with the policy effect, $R_{\mathrm{occ}} = +0.006\,[-0.018, +0.029]$ over 342 units, though no equivalence margin was specified.

Two airline production-serving attempts on the same 50 tasks shared a separate policy-effect campaign (Appendix W.1). The first, after recovery of 40 timed-out units, gave $R_{\mathrm{occ}} = -0.041\,[-0.125, +0.041]$ on 123 of 150 units, crossing the ± 0.10 boundary, and its recovery departed from the registered count-only review. A technical repeat gave $+0.007\,[-0.060, +0.080]$, within the band, but its within-campaign residual, $-0.027\,[-0.053, -0.007]$, and four of 273 disagreeing adjacent comparisons show that the band result alone does not validate the clones. Under

deterministic serving no clone comparison changed a reward, consistent with serving variation as the source, although the original four cannot be replayed. There, the noticed-refusal residual against separately started whole episodes was $-0.020\,[-0.100, +0.040]$. The sum equaled the effect on 47 of 50 tasks, including 35 of 37 with visits (Figure 3(d)). The other three started with empty opening messages (Appendix Z.3). For the reviewed engine, which charges each review, the sum equaled the effect on all 50 tasks, $R_{\text{occ}} = 0.000\,[0.000, 0.000]$, while the count-scaled first value fell short, $-0.040\,[-0.100, 0.000]$. Reviewed allows had zero value at all 51 visits, and one of seven budget-exhausting reviews changed the outcome (Appendix Z.4).

6.4 Guard rankings with a second actor

With Qwen2.5-7B as actor and customer on 50 airline tasks under deterministic serving, the registered silent test replicates. One silent refusal has zero value at all 44 branch points, with a gap of $+0.227\,[+0.114, +0.364]$ (Figure 2(a), Appendix Z.6). Whole episodes give both refusals $+0.20\,[+0.10, +0.32]$ and the engine $+0.02\,[0.00, +0.06]$. First-point and count-scaled values rank the engine above silent refusal (Figure 2(f)), whereas noticed-refusal and engine occupancy sums equal each task's measured effect (Figure 3(d)). With notice, 22 of 165 visits have nonzero values, of either sign within one walk, summing to each task's effect (Figure 3(b)). In 1,000 post hoc random splits, guards are chosen and scored on separate task halves. First-point values select worse guards in half the choices, with mean held-out regret 0.107, and count-scaled values in a fifth, with 0.049, whereas occupancy sums (among walked guards only) and whole episodes never do (Figure 2(g) and (h), Appendix Z.6). Whole episodes select the guard with 288 thousand actor tokens per task versus 899 thousand for walks. Walks locate guard effects (Figure 3(c)) and test cheap local substitutes for policy effects.

7 Limitations

We neither reproduce SABER's full system (Cuadron et al., 2025) nor tune arms. Reference-mutation labels denote use rather than necessity, and TypeBlock's matching taxonomy partly defines the sign flip. Budgets cap slots, not tokens, and one verifier arm spends four slots per call. The 14B and 32B actors each solve 21 of 114 retail tasks, with 11 shared. Branch experiments were not repeated at 32B, and the customer model was held fixed in the 32B probe (Appendix X.1). Most branch points cannot show a difference, 81.1% in retail and 53.0% in airline (Appendix X). Greedy-decoding seeds repeat tasks, retail occupancy ran once, and both airline production attempts reuse one task pool and policy-effect campaign. The fixed silent schedule loses 17 last points with outcome-dependent missingness, and full silent-refusal coverage requires 3.8 times as many clones (Appendix R.1). Deterministic checks use one seed, at most 50 tasks and three serving stacks. The airline-only second-actor study replaces actor and simulated customer together, so it does not isolate actor transfer. Benchmark thread timing permits reset divergence between separately started episodes and rarely changes continuation observations. Audit and branch intervals concern different populations. Branch estimates select mutating proposals and assign zero where the guard never fires, and the registered first-point tests concern silent refusal. Guard selection was tested only post hoc on one pool and the instrument only on τ^2-bench, so conclusions are limited to the tested actors, budgets and interventions.

8 Conclusion

At matched budget, blanket type-blocking helps the no-mutation stratum and hurts the mutation stratum. Silent first-point substitution fails in both domains and for a second actor, and first-point values can misorder guards. With every visit measured, deterministic serving and synchronized starts, the occupancy sum equals the policy effect task by task, while a fixed visit schedule and production-serving campaigns leave it unresolved. First-point values should not be reported as policy effects, and intervention values should state their support.

AI USE STATEMENT

We used large language models for code implementation assistance, manuscript copy-editing and polishing, conceptual schematic icons, and discussion during study design. All experimental workflows, data analyses and scientific claims were designed, conducted and verified by the authors.

ETHICS STATEMENT

The experiments use the public τ^2-bench and AgentDojo task suites, in which agents act on simulated environments and receive requests from a simulated user or a fixed task prompt. No human subjects took part and no personal data were collected. The released records contain only conversations and tool calls generated inside these environments. The finding that blanket type-blocking lowers task success on reference-mutation tasks is not evidence that guards are unnecessary. The same guard removes mutation-associated failures by construction, and whether to deploy it depends on the weight placed on irreversible actions (Appendix G). GuardValue executes proposed actions inside cloned episodes. Applied to an agent whose tools reach real users or external services, a clone would execute those actions for real, so the instrument should run only in sandboxed or simulated environments.

REPRODUCIBILITY STATEMENT

The supplementary artifact provides de-identified records and offline analysis commands. Appendices O and Z describe the environments and measurement procedure. Appendix R states the decision rules. The artifact documents which results can be recomputed without a served model and which require the live benchmark. Appendix Z.2 documents the serving-determinism diagnostic and its records.

REFERENCES

Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. Safe reinforcement learning via shielding. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, volume 32, 2018. doi: 10.1609/aaai.v32i1.11797. URL https://ojs.aaai.org/index.php/AAAI/article/view/11797.

Rahul Suresh Babu and Laxmipriya Ganesh Iyer. Contract2Tool: Learning preconditions and effects for reliable tool-augmented LLM agents, 2026. URL https://arxiv.org/abs/2606.07904.

Arko Banerjee, Kia Rahmani, Joydeep Biswas, and Isil Dillig. Dynamic model predictive shielding for provably safe reinforcement learning. In *Advances in Neural Information Processing Systems 37*, 2024. URL https://papers.neurips.cc/paper_files/paper/2024/hash/b589d92785e39486e978fa273d0dc343-Abstract-Conference.html.

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2-bench: Evaluating conversational agents in a dual-control environment, 2025. URL https://arxiv.org/abs/2506.07982.

Akash Bonagiri, Devang Borkar, Gerard Janno Anderias, Setareh Rafatirad, and Houman Homayoun. CausalFlow: Causal attribution and counterfactual repair for LLM agent failures, 2026. URL https://arxiv.org/abs/2605.25338.

Alejandro Cuadron, Pengfei Yu, Yang Liu, and Arpit Gupta. SABER: Small actions, big errors—safeguarding mutating steps in LLM agents, 2025. URL https://arxiv.org/abs/2512.07850.

Murad Dawood, Ahmed Shokry, and Maren Bennewitz. A dynamic safety shield for safe and efficient reinforcement learning of navigation tasks. In *Proceedings of the 7th Annual Learning for Dynamics and Control Conference (L4DC)*, volume 283 of *PMLR*, pp. 686–697, 2025. URL https://proceedings.mlr.press/v283/dawood25a.html.

Edoardo Debenedetti, Jie Zhang, Mislav Balunović, Luca Beurer-Kellner, Marc Fischer, and Florian Tramèr. AgentDojo: A dynamic environment to evaluate prompt injection attacks and defenses for LLM agents. In *Advances in Neural Information Processing Systems 37, Datasets and Benchmarks Track*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/97091a5177d8dc64b1da8bf3e1f6fb54-Abstract-Datasets_and_Benchmarks_Track.html.

Sajib Acharjee Dip, Dawei Zhou, and Liqing Zhang. Think again or think longer? Selective verification for budget-aware reasoning, 2026. URL https://arxiv.org/abs/2606.19808.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. CRITIC: Large language models can self-correct with tool-interactive critiquing. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/fef126561bbf9d4467dbb8d27334b8fe-Abstract-Conference.html.

Audrey Huang, Adam Block, Qinghua Liu, Nan Jiang, Akshay Krishnamurthy, and Dylan J. Foster. Is Best-of-N the best of them? Coverage, scaling, and optimality in inference-time alignment. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 25075–25126, 2025. URL https://proceedings.mlr.press/v267/huang25c.html.

Nils Jansen, Bettina Könighofer, Sebastian Junges, Alex Serban, and Roderick Bloem. Safe reinforcement learning using probabilistic shields. In *31st International Conference on Concurrency Theory (CONCUR 2020)*, volume 171 of *LIPIcs*, 2020. URL https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.CONCUR.2020.3.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/edac78c3e300629acfe6cbe9ca88fb84-Abstract-Conference.html.

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pp. 267–274, 2002.

Bettina Könighofer, Julian Rudolf, Alexander Palmisano, Martin Tappler, and Roderick Bloem. Online shielding for reinforcement learning. *Innovations in Systems and Software Engineering*, 19(4), 2023. doi: 10.1007/s11334-022-00480-4. URL https://arxiv.org/abs/2212.01861.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 611–626, 2023. doi: 10.1145/3600006.3613165. URL https://doi.org/10.1145/3600006.3613165.

Will LeVine, Brendan Evers, Sam Saltwick, and Abhay Venkatesh. RubricRefine: Improving tool-use agent reliability with training-free pre-execution refinement, 2026. URL https://arxiv.org/abs/2605.09730.

Daoyuan Li. Safe under budget? Verification budgets and abstention failures in web agents. In *Second Workshop on Agents in the Wild: Safety, Security, and Beyond (AIWILD), ICML*, 2026.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. AWQ: Activation-aware weight quantization for on-device LLM compression and acceleration. In *Proceedings of Machine Learning and Systems*, volume 6, 2024. URL https://proceedings.mlsys.org/paper_files/paper/2024/hash/42a452cbafa9dd64e9ba4aa95cc1ef21-Abstract-Conference.html.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-Refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems 36*, 2023. URL https://papers.nips.cc/paper_files/paper/2023/hash/91edff07232fb1b55a505a9e9f6c0ff3-Abstract-Conference.html.

Dipkumar Patel. Anytime verified agents: Adaptive compute allocation for reliable LLM reasoning under budget constraints. *Transactions on Machine Learning Research*, 2026. URL https://openreview.net/forum?id=JMDCMf7mlF.

Hubert M. Pyskło, Artem Zhuravel, and Patrick D. Watson. Agent-Diff: Benchmarking LLM agents on enterprise API tasks via code execution with state-diff-based evaluation. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Volume 2*, pp. 9707–9717, 2026. doi: 10.1145/3770855.3817555. URL https://doi.org/10.1145/3770855.3817555.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Yeonju Ro, Haoran Qiu, Íñigo Goiri, Rodrigo Fonseca, Ricardo Bianchini, Aditya Akella, Zhangyang Wang, Mattan Erez, and Esha Choukse. Sherlock: Reliable and efficient agentic workflow execution, 2025. URL https://arxiv.org/abs/2511.00330.

Fabio Rovai. CIVeX: Causal intervention verification for language agents, 2026. URL https://arxiv.org/abs/2605.09168.

Tanmay Sah, Vishal Srivastava, Dolly Sah, and Kayden Jordan. The verifier tax: Horizon dependent safety–success tradeoffs in tool using LLM agents. In *Proceedings of the ACM Conference on AI and Agentic Systems*, pp. 785–799, 2026. doi: 10.1145/3786335.3813160. URL https://doi.org/10.1145/3786335.3813160.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems 36*, pp. 68539–68551, 2023. doi: 10.52202/075280-2997. URL https://proceedings.neurips.cc/paper/2023/hash/d842425e4bf79ba039352da0f658a906-Abstract-Conference.html.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1889–1897, 2015. URL https://proceedings.mlr.press/v37/schulman15.html.

Jaineet Shah. Causal agent replay: Counterfactual attribution for LLM-agent failures, 2026. URL https://arxiv.org/abs/2606.08275.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems 36*, 2023. URL https://papers.nips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/1b623663fd9b874366f3ce019fdfdd44-Abstract-Conference.html.

Yueh Tang and Justin Zhan. Verify what matters: Budgeted verification for tool-using agents under counterfactual downstream harm. *Transactions on Machine Learning Research*, 2026. URL https://openreview.net/forum?id=nv1jzr0FaZ.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. AppWorld: A controllable world of apps and people for benchmarking interactive coding agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16022–16076, 2024. doi: 10.18653/v1/2024.acl-long.850. URL https://aclanthology.org/2024.acl-long.850/.

Haoyu Wang, Christopher M. Poskitt, and Jun Sun. AgentSpec: Customizable runtime enforcement for safe and reliable LLM agents. In *Proceedings of the 2026 IEEE/ACM 48th International Conference on Software Engineering*, pp. 2938–2950, 2026. doi: 10.1145/3744916.3764546. URL https://doi.org/10.1145/3744916.3764546.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=roNSXZpUDN.

Aojie Yuan, Zhiyuan Su, and Yue Zhao. AEGIS: No tool call left unchecked—a pre-execution firewall and audit layer for AI agents, 2026. URL https://arxiv.org/abs/2603.12621.

Guibin Zhang, Junhao Wang, Junjie Chen, Wangchunshu Zhou, Kun Wang, and Shuicheng Yan. AgenTracer: Who is inducing failure in the LLM agentic systems? In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/134ed7a477770f227f12450ef0cbb8f4-Abstract-Conference.html.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. WebArena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/4410c0711e9154a7a2d26f9b3816d1ef-Abstract-Conference.html.

Xiaolin Zhou, Jinbo Liu, Li Li, Ryan A. Rossi, and Xiyang Hu. Counterfactual trace auditing of LLM agent skills, 2026. URL https://arxiv.org/abs/2605.11946.

APPENDIX

The appendix is ordered by the claim each part supports. Appendices A–N give the matched-budget audit behind Section 3, from arm specifications to strata and transfer. Appendices O–R describe the environments, the branching procedure, the two budgets and the registered decision rules behind Sections 5 and 6. Appendices S–W report each branch study, including the early short-horizon runs that are excluded from every claim and both airline occupancy attempts. Appendices X–Z cover the measurement floor, every deviation and correction, and clone conformance with reproduction details. The appendix usually reports estimates at full precision, and the main text rounds them.

CONTENTS

A Audit Figures

B Matched-Budget Runner

C Arm Specifications

D NecessityGate Specification

E ReflectRepair Specification and Full Results

F Guard-Failure Decomposition

G Deployment Contrasts

H Per-Domain Regime Composition

I Capability and the Sign Flip

J Cross-Fitted Arm Choice

K Irreversible-Taxonomy Sensitivity

L Capability-Conditioned Guard Value

M Capability Axis

N Cross-Benchmark Transfer

O Reproducibility Details

P Branching a Live Benchmark Episode

Q Separate Interaction and Intervention Budgets

R Estimators and the Pre-Registration Record

S Excluded Runs of the First, Truncated Instrument

T Experiment 4: Whole-Episode Policies

U Experiment 6: One Refusal or One Policy, From the Same Prefix

V Experiment 7: A Second Domain

W Experiment 8: The Identity Along the Guarded Policy's Own Visits

X The Measurement Floor

Y Deviations and Corrections

Z Clone Conformance and Reproducibility of the Branch Experiments

A AUDIT FIGURES

Figure A.1 summarizes the confirmatory contrasts, the domain effects, the two deployment axes and the AgentDojo transfer with intervals. Figure A.2 adds the success-risk frontier, the break-even risk weights and a leave-one-seed-out check on point estimates. The headline contrasts are stated in Section 3, and Tables G.1, H.1, L.1 and N.1 give the remaining values.

B MATCHED-BUDGET RUNNER

The per-episode runner is Algorithm B.1. Each arm receives a fresh budget meter that records all its intervention charges. An action proposed after the cap returns a budget-exhausted failure at the prior state.

The runner records success, mutation-associated failure and budget used per episode. The per-arm TSR and MAF are means over episodes, and the deployment utility of arm a is the aggregate $\widehat{U}_a(\lambda,\gamma) = \widehat{\mathrm{TSR}}_a - \lambda\widehat{\mathrm{MAF}}_a - \gamma\,\bar{c}_a/B$, with risk weight λ, slot-cost weight γ and mean slots used $\bar{c}_a$. It is not a per-episode quantity, and λ^* is the risk weight at which an arm's utility equals NoGuard's.

C ARM SPECIFICATIONS

Each arm specifies a decision rule for a proposed action and a charge to the shared per-episode meter. Table C.1 states both. Every arm sees the transcript, the tool schemas and the proposed call.

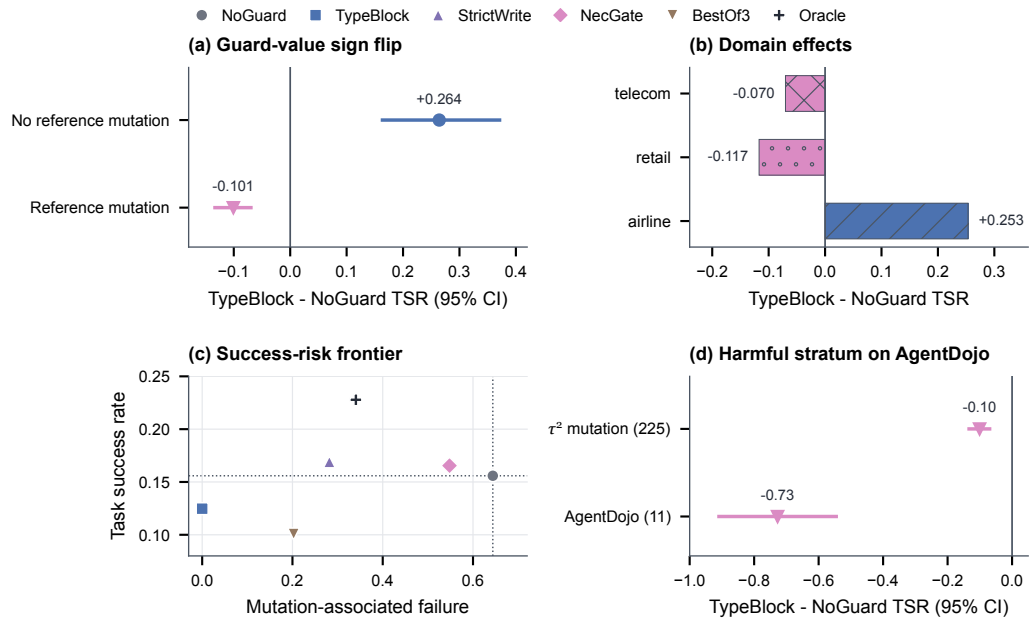

Figure A.1: **The matched-budget audit reveals conditional guard value. (a)** Confirmatory TypeBlock contrasts by reference-mutation stratum. **(b)** Domain effects reflect different task mixtures. **(c)** Task success and mutation-associated failure are separate deployment axes. **(d)** The harmful stratum transfers to 11 capable AgentDojo tasks whose gold trajectory uses a mutation tool. Panels (a) to (c) use 278 τ^2-bench tasks and panel (d) 11 capable AgentDojo tasks, at matched budget $B = 10$ with three seeds. Intervals are task-clustered 95% bootstrap intervals. Positive effects are blue and negative effects pink. Bars start at zero, hatching provides a second visual cue, and marker shapes distinguish arms.

Algorithm B.1 Matched-Budget Runner

Require: episode e, budget B, arms $\mathcal{A}$, backbone M
Ensure: per-episode record table T

$T \leftarrow \emptyset$
for all $a \in \mathcal{A}$ **do**
  $m \leftarrow \textsc{BudgetMeter}(B)$
  while arm a proposes a step (guard check, replan, confirmation, candidate, verifier call, or tool) **do**
    if charging the step would exceed m **then**
      stop with a budget-exhausted failure at the state before the step
    else
      charge the step to m and execute it
    end if
  end while
  record in $T[a]$: success, mutation-associated failure and budget used
end for
return T

REPLANONCE and BESTOF3ACTIONVERIFIER additionally receive the pre-action context object, which carries the task identifier and an irreversibility flag but no gold trajectory and no outcome. No arm except ORACLE sees a reference trajectory or any evaluation outcome, so no deployable arm can route on the answer it is being scored against. The task identifier is an opaque key, not an outcome, and REPLANONCE uses it only to derive a reproducible seed.

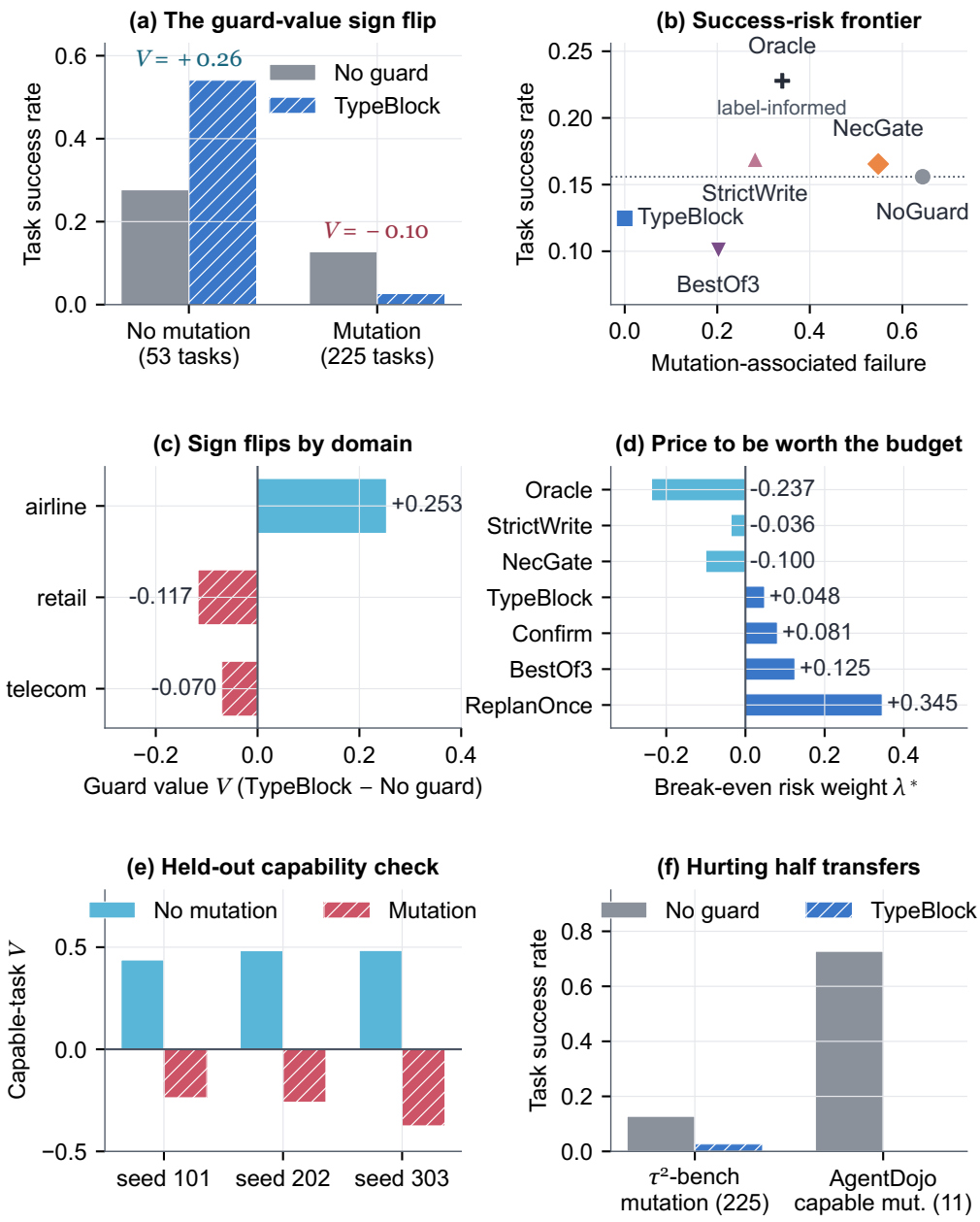

Figure A.2: **Guard value varies with the reference-mutation mixture rather than having a single sign.** Matched budget $B = 10$, three seeds, 278 τ^2-bench tasks, point estimates throughout. The stratum intervals are in Table 1. (a) TypeBlock takes opposite signs on the no-mutation ($V = +0.264$) and mutation ($V = -0.101$) strata, and their intervals are disjoint (Table 1). (b) Risk increases from left to right, so the preferred frontier is up and to the left. StrictWrite has higher success and lower risk than NoGuard, while TypeBlock trades success for zero mutation-associated failure. Oracle is label-informed and not a deployable arm. (c) The sign flips by domain. (d) The break-even risk weight λ^*, the risk weight at which an arm's utility equals NoGuard's, at slot-cost weight $\gamma = 0$ and on point estimates. A negative value means the arm already exceeds NoGuard at $\lambda = 0$. The zero-overhead variant shares TypeBlock's weight and is omitted, as is ReflectRepair, whose weight is zero (Table G.1). (e) A leave-one-seed-out check keeps the flip in every fold. The three seeds re-run one pool under greedy decoding, not three independent samples. (f) The negative contrast replicates on AgentDojo. Its 11 tasks were selected for capability and use a mutation tool in the gold trajectory, while the 225 τ^2-bench tasks are the unselected reference-mutation stratum, so denominators accompany both pairs of bars, and their magnitudes are not compared.

Table C.1: The ten arms as decision rules and their charges. Every intervention is charged to the same per-episode meter as its proposed action. Each arm therefore shares the reference arm's interaction cap. TypeBlock-ZeroOverhead and TypeBlock-StrictWrite are sensitivities. The first separates TypeBlock's blocking effect from its overhead. The second tests whether the effect persists after removing the one boundary tool from the taxonomy.

Arm	Decision rule on a proposed action	Charged to the matched meter
NoGuard	Execute.	The action.
TypeBlock	Block iff the tool name is in the irreversibility taxonomy. Deterministic, so precision and recall against that taxonomy are both one.	One decision slot per checked action.
TypeBlock-ZeroOverhead	Identical rule, with the decision slot waived.	The action only.
TypeBlock-StrictWrite	Identical rule over the taxonomy minus the human-escalation operation.	One decision slot per checked action.
NecessityGate	One language-model call per irreversible action returns ALLOW or BLOCK. Block-precedence parsing, and any ambiguous, empty or failed reply fails safe to BLOCK and is counted.	One decision slot and the decision's tokens per irreversible action.
ReflectRepair	One language-model call per irreversible action returns a three-part reply, a one-line reflection, an ALLOW or BLOCK decision, and on a block a one-sentence repair hint returned to the agent through the tool channel so it can revise rather than only learn it was blocked.	As NecessityGate, plus the hint's tokens.
ReplanOnce	On the first irreversible proposal of an episode, re-query the actor once from the same untouched pre-action history under an independent derived seed, then execute whatever the replan proposes. Never fires twice, and abstains rather than starting a replan it cannot also afford to execute.	One slot for the replan call and its tokens, plus the executed action.
ConfirmBeforeMutate	Ask a simulated user, given the transcript and the exact call and arguments, to reply CONFIRM or DENY. Execute only on an explicit confirmation, otherwise block.	One slot and the decision's tokens per mutation.
BestOf3ActionVerifier	Sample three candidate next actions and let a verifier pick one before anything executes. The verifier reads only the pre-action history and is instructed never to assume an action succeeded.	Two slots for the extra candidates, one for the verifier, plus their tokens and the executed action.
Oracle	Label-informed reference, not a deployable policy. It allows an irreversible action iff the reference trajectory uses that tool, and blocks it otherwise.	One decision slot per checked action, like TypeBlock, plus the executed action.

Two arms differ from the methods whose names they resemble. REPLANONCE is not a retry loop, because a deployed agent cannot retry a mutation that already executed. The arm instead spends its budget on one independent reconsideration of a pending action. This replanning remains available before an irreversible action executes. BESTOF3ACTIONVERIFIER selects among candidate actions before execution rather than among completed trajectories. Selecting among completed trajectories would require executing every candidate or cloning the environment. The former multiplies the irreversible actions being measured, and a deployed agent does not have an environment clone.

Observed baseline behavior. Episode records and per-step traces quantify each deployable baseline's activity and help interpret its performance and budget use. Extra proposals were sampled at temperature 0.2 under independently derived seeds, so a repeated call is the actor agreeing with itself rather than greedy decoding. BESTOF3ACTIONVERIFIER wraps every tool call, not only mutations, and one selection costs four slots, so at $B = 10$ it can afford two. It executes exactly two tool calls in 732 of its 834 episodes. In 768 episodes, it can no longer afford a selection and answers

in text. Its candidates are also rarely distinct. All three candidates name the same tool in 1,363 of 1,628 selections. The verifier keeps the actor's own proposal in 1,041, and 420 of the 522 executed selections of another candidate are identical to the original call.

REPLANONCE fired 662 times. The replan answered in text 185 times, and of the 477 replans that executed, 347 repeated the original call exactly, 21 changed only its arguments and 109 chose another tool. NECESSITYGATE never produced an unparseable reply (Appendix D).

Its audit result therefore reflects a four-slot charge for every call, not verification in general. Against NoGuard its task-micro success gap is $-0.055\,[-0.088, -0.024]$ and REPLANONCE's is $-0.023\,[-0.043, -0.004]$, both excluding zero and both negative, so neither improves task-micro success at $B = 10$. This result concerns these arms at this budget, not best-of-N selection in general.

No arm was tuned. Every prompt, threshold, temperature, candidate count and parsing rule was fixed in the protocol of 17 July 2026, before the primary run, and none was subsequently changed for any arm, including the guards. The comparisons therefore concern pre-specified rather than optimized arms. A tuned version of any arm could score differently.

TYPEBLOCK-ZEROOVERHEAD separates two sources of guard cost. A guard can block a needed action, or it can consume needed budget. Waiving the decision slot removes the second, so any residual gap between it and TYPEBLOCK is the overhead and any gap that remains against NOGUARD is the blocking itself.

D NECESSITYGATE SPECIFICATION

NecessityGate is the served backbone (Qwen2.5-14B-Instruct-AWQ) queried at temperature 0 with a 16-token cap. For each proposed irreversible action it sends the rendered conversation so far, fenced as untrusted data. The proposed action's tool name and arguments follow outside the fence. The instruction is "Reply with exactly one word: ALLOW or BLOCK." The parser gives blocking responses precedence. Any of BLOCK, DISALLOW, DENY, or a refusal phrase yields BLOCK, a standalone ALLOW yields ALLOW, and any ambiguous, empty or failed reply fails safe to BLOCK. The budget meter charges each NecessityGate query to the same matched budget as a tool call. In the reported 14B audit, NecessityGate made 1,210 queries and returned 377 blocks. No reply failed to parse and no call failed. Replies used 2.0 completion tokens on average, and the 16-token cap was never binding.

E REFLECTREPAIR SPECIFICATION AND FULL RESULTS

ReflectRepair keeps NecessityGate's per-call necessity judgment and adds a repair step of our own design. On a block it returns a one-sentence repair hint through the tool channel. The agent can then revise the blocked action rather than only learn that it was blocked. It is charged one decision slot per irreversible action, as NecessityGate is. Its reflection tokens are metered as a separate cost dimension. It runs alongside the other nine arms in the primary session, with the same tasks, seeds and serving conditions.

Table E.1 gives the per-stratum task success. Over all 278 tasks ReflectRepair reaches 0.156 TSR, the same rounded estimate as NoGuard (0.156), with an equal-domain gap of $+0.020\,[-0.007, +0.047]$, so no difference is detected. It uses 6.78 interaction slots on average, with its mutation-associated failure rate still at 0.501. In the strata it reaches 0.390 on no-mutation tasks against TypeBlock's 0.541, and 0.101 on mutation tasks against the unguarded 0.127. These runs detect no task-success advantage from adding repair to NecessityGate and do not isolate the hint's effect.

F GUARD-FAILURE DECOMPOSITION

TypeBlock executes zero irreversible actions by construction, so its mutation-associated failure rate is exactly zero. The reference-mutation stratum depends on the reference trajectory's tool use, but

Table E.1: Task success per stratum and over all tasks (All TSR), and mutation-associated failure rate (All MAF) on the 14B primary run at $B = 10$, the per-episode interaction cap, three seeds over the complete pool of 278 tasks (53 no-mutation, 225 mutation). NecGate abbreviates NecessityGate. Paired gaps against NoGuard differ under the two reported weightings. NecessityGate's equal-domain macro gap is $+0.023\,[+0.002, +0.045]$ and its task-micro gap is $+0.010\,[-0.010, +0.029]$, while ReflectRepair's are $+0.020\,[-0.007, +0.047]$ and $-0.000\,[-0.022, +0.022]$. The label-informed Oracle differs from NoGuard under both weightings. ReflectRepair matches NoGuard's rounded task success with a mean of 6.78 interaction slots, and its success on no-mutation tasks is 0.390 against TypeBlock's 0.541.

Stratum	NoGuard	TypeBlock	NecGate	ReflectRepair	Oracle
No mutation (53)	0.277	0.541	0.352	0.390	0.541
Mutation (225)	0.127	0.027	0.121	0.101	0.154
All TSR (278)	0.156	0.125	0.165	0.156	0.228
All MAF (278)	0.644	**0.000**	0.548	0.501	0.341

Table F.1: Successful episodes per arm, with each stratum split by whether the scorer's reference-action replay changes the database used for comparison. The replay calls no model. These counts were computed after the audit's registered decision and have no interval.

stratum	the reference replay	tasks	TypeBlock	NoGuard	Oracle
reference mutation	changes the scored database	199	0 of 597	53 of 597	51 of 597
reference mutation	leaves it unchanged	26	18 of 78	33 of 78	53 of 78
no mutation	changes the scored database	23	0 of 69	1 of 69	1 of 69
no mutation	leaves it unchanged	30	86 of 90	43 of 90	85 of 90

the benchmark's scorer does not inspect which tools the agent called. Under its default reward basis, the scorer replays reference actions on a fresh copy of the initial state, skipping any that raise an exception. It requires the agent's final database to match. We replayed every task's reference in exactly that way, with no model involved, and split each stratum by whether the replay changes the scored database (Table F.1).

On the 199 reference-mutation tasks whose reference changes the scored database, TypeBlock succeeds in none of 597 episodes. The other 26 leave the database unchanged. In 24 tasks, the reference uses only one taxonomy tool, transfer to a human agent, which writes nothing. In 2 tasks, the reference's exchange call raises an exception during scorer replay. All 18 of TypeBlock's successes on the stratum, six tasks under every seed, fall on those 26. Its 0.027 therefore comes from the scorer accepting an unchanged database where the reference changes nothing the scorer compares. Conversely, the 23 no-mutation tasks whose reference changes the scored database are all telecom tasks. Their references modify only the customer's device through customer-side tools outside the agent taxonomy (Appendix K), and every arm sits at the floor there. The harm on the reference-mutation stratum is the block, not the cost of the check. TypeBlock uses fewer interaction slots than the semantic-review arms, yet it never succeeds where the scored state requires a write.

G Deployment Contrasts

Figure G.1 shows equal-domain, task-clustered paired bootstrap differences between each guard policy and NoGuard in task success and in mutation-associated failure over the complete pool of 278 tasks. For task success, StrictWrite's interval excludes zero ($+0.057\,[+0.013, +0.101]$), as does NecessityGate's marginally ($+0.023\,[+0.002, +0.045]$). TypeBlock's and ReflectRepair's intervals straddle zero, and BestOf3ActionVerifier has a negative point estimate. Every policy has lower mutation-associated failure than NoGuard, with TypeBlock lowest at zero. The break-even weight in Table G.1 summarizes these arm-specific success changes against the shared reduction in risk.

Table G.1: All-task intention-to-treat over the 278 tasks, matched $B = 10$, three seeds, point estimates. Slots denotes mean interaction slots used and Tokens mean tokens per episode, which the cap does not equalize, and λ^* is the break-even risk weight at which the arm and NoGuard reach equal utility $U = \text{TSR} - \lambda\,\text{MAF}$ (task-clustered, $\gamma = 0$). A negative λ^* means the arm's point-estimate utility already exceeds NoGuard at $\lambda = 0$. Arm comparisons other than the two stratum contrasts are exploratory.

Arm	TSR	MAF	Slots	Tokens	λ^*
NoGuard	0.156	0.644	5.91	56.9k	—
TypeBlock	0.125	**0.000**	6.20	64.9k	0.048
TypeBlock-ZeroOvhd	0.125	**0.000**	**4.68**	66.7k	0.048
StrictWrite	**0.169**	0.282	5.79	57.9k	−0.036
NecessityGate	0.165	0.548	6.73	56.1k	−0.100
ReflectRepair	0.156	0.501	6.78	61.1k	0.000
ReplanOnce	0.133	0.578	6.39	58.1k	0.345
ConfirmBeforeMutate	0.146	0.525	6.59	57.6k	0.081
BestOf3ActionVerifier	0.101	0.203	7.73	65.6k	0.125
Oracle (label-informed)	0.228	0.341	6.76	57.2k	−0.237

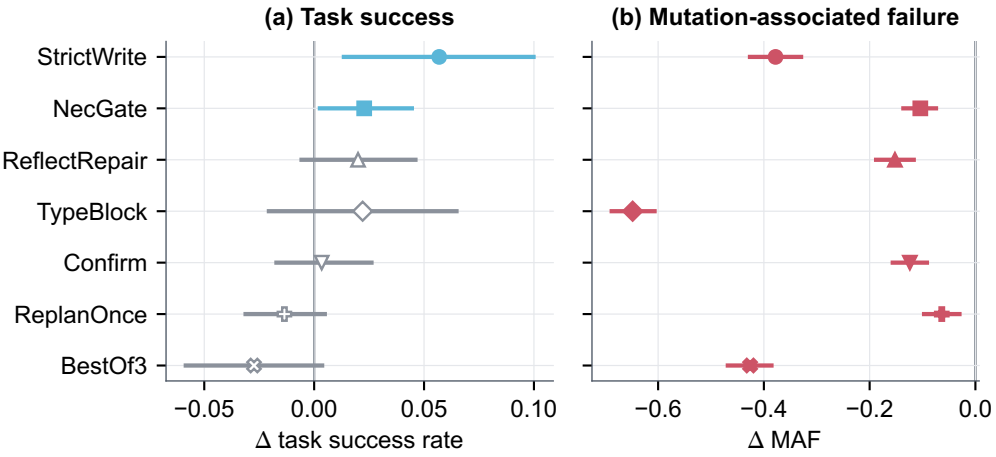

Figure G.1: Paired guard-policy differences (policy minus NoGuard) under equal-domain macro weighting, with task-clustered bootstrap 95% intervals over all 278 tasks. Table G.1 reports the same arms task-micro, and the two weightings disagree for NecessityGate and ReflectRepair. The left panel is the change in task success rate and the right the change in mutation-associated failure rate. StrictWrite and NecessityGate raise task success with intervals excluding zero, while TypeBlock, ReflectRepair and BestOf3 do not. Every policy lowers mutation-associated failure.

H Per-Domain Regime Composition

Table H.1 shows the reference-mutation mixture and the guard-value sign per domain over the complete pools. Airline is the only domain where no-mutation tasks are common and the agent solves some of them, and type-blocking helps there ($V = +0.253$), which is consistent with that mixture. Retail is almost entirely mutation tasks, so the guard blocks tools the reference uses and hurts ($V = -0.117$). Telecom is mutation-heavy and the 14B agent's success is low for every arm, and type-blocking lowers it further ($V = -0.070$). The positive side of the sign flip thus rests on airline's mixture and capability, and the negative side on the mutation-heavy retail and telecom pools.

Table H.1: Per-domain composition and guard value for Qwen2.5-14B at $B = 10$ over the complete pools. No mutation and Mutation are the task counts by reference-trajectory mutation stratum, and $V = \text{TSR(TypeBlock)} - \text{TSR(NoGuard)}$ is the micro-averaged guard value in that domain.

Domain	Tasks	No mutation	Mutation	V
Airline	50	23	27	$+0.253$
Retail	114	7	107	-0.117
Telecom	114	23	91	-0.070

Table J.1: Cross-fitted choice between TypeBlock and NecessityGate over the 278 tasks, three folds, under the fixed weights $\lambda = 1.0$ and $\gamma = 0.05$, point estimates. Both variants select TypeBlock in every fold, matching blanket TypeBlock and trailing NoGuard, and capture none of the oracle's headroom. No task feature enters the choice, and NoGuard is not a candidate. λ^* is the break-even risk weight against NoGuard.

Policy	TSR	Δ vs NoGuard	λ^*
NoGuard	0.156	—	—
CrossFit best-constant	0.125	-0.031	0.048
CrossFit per-domain	0.125	-0.031	0.048
Oracle (label-informed)	0.228	$+0.072$	-0.237

I Capability and the Sign Flip

The positive side of the sign flip requires no-mutation tasks that the agent can solve, and a positive aggregate domain effect occurs only on airline in these pools. Telecom's no-mutation tasks are rarely solved by any arm, including the oracle (Table F.1), and retail has few no-mutation tasks (Table H.1). The confinement of the positive domain effect to airline is consistent with both agent capability and the benchmark's task mixture, which these data do not separate.

J Cross-Fitted Arm Choice

We test whether choosing a guard policy without oracle labels recovers the performance advantage those labels reveal. The choice is cross-fitted over the 278 tasks on fixed folds. This choice is narrower than feature-based routing. In each fold it takes the arm with the higher risk-weighted training utility from two candidates, TypeBlock and NecessityGate, under the fixed weights $\gamma = 0.05$ and $\lambda = 1.0$. NoGuard is not a candidate, and no task feature enters the choice. It selects TypeBlock in all three folds. The per-domain variant also selects TypeBlock in every fold and domain.

Table J.1 gives the result. Both variants select the same constant guard policy. Each scores 0.125 TSR, identical to blanket TypeBlock and below NoGuard at 0.156, with a break-even risk weight of 0.048, and neither captures any of the headroom a label-informed oracle reaches at 0.228. Feature-based routing was not evaluated in the reported experiments.

K Irreversible-Taxonomy Sensitivity

The irreversible-action taxonomy is every agent tool that τ^2-bench marks as state-changing, plus the human-escalation operation as its single boundary case. It is limited to the agent's tools because those are the only calls a guard sees. Telecom also gives the simulated customer tools that change the customer's own device, such as toggling airplane mode, and these lie outside the taxonomy. Mutation, the reference-mutation strata and mutation-associated failure therefore all refer to agent-side writes. The TypeBlock-StrictWrite arm, StrictWrite in the main text, applies TypeBlock after removing that boundary tool from the taxonomy. StrictWrite was run as a full arm, not derived by post hoc relabeling, and therefore provides the taxonomy sensitivity analysis. Narrowing the taxonomy keeps the sign flip. StrictWrite matches TypeBlock on no-mutation tasks (0.541) and still trails NoGuard on mutation tasks (0.081 against 0.127), so both positive and negative effects persist

Table L.1: Capability-conditioned guard value with separate seeds for capability definition and scoring. For each held-out seed a task is capable when the NoGuard or oracle arm succeeds under either other seed. The guard value $V = \text{TSR}(\text{TypeBlock}) - \text{TSR}(\text{NoGuard})$ is then scored on the held seed's capable tasks, split by mutation stratum. The sign flip holds in every fold.

	No mutation		Mutation	
Held seed	V	n	V	n
101	$+0.438$	32	-0.237	59
202	$+0.483$	29	-0.259	54
303	$+0.484$	31	-0.375	56

under this boundary choice. StrictWrite's all-task point estimate is also higher than NoGuard's (0.169 against 0.156), a difference the audit does not resolve.

L Capability-Conditioned Guard Value

The primary estimand is the all-task intention-to-treat. For the capability-conditioned secondary analysis, we avoid defining capability on the episodes used for scoring, which would be circular. We hold out one seed at a time and call a task capable when the NoGuard or oracle arm succeeds on it under either of the other two seeds. We then score TypeBlock minus NoGuard on the held seed's capable tasks. Table L.1 reports each held seed and stratum. The sign of the flip persists across all three folds. On capable no-mutation tasks TypeBlock helps by roughly $+0.47$, and on capable mutation tasks it hurts by roughly -0.29.

M Capability Axis

The capability axis is visible within the primary run through the per-domain floor (Tables H.1 and F.1), where telecom's low success leaves little room for guarding, and through the leave-one-seed-out capability conditioning (Table L.1). A Qwen2.5-7B replication of the full ten-arm protocol on the same 278-task pools directly tests this capability comparison. It changes the backbone while holding the tasks, seeds and budget fixed.

On the 7B backbone the sign flip reproduces with the same structure as the 14B strata in Table 1. TypeBlock lifts success from 0.31 to 0.52 on the no-mutation stratum, a guard value of $+0.214\,[+0.119, +0.321]$. It drops success from 0.12 to 0.03 on the mutation stratum, $-0.095\,[-0.135, -0.058]$. Both intervals are well separated from zero. As with the 14B actor, the pooled effect over all 278 tasks is unresolved, $-0.036\,[-0.076, +0.004]$, while mutation-associated failure again falls to zero. Airline still carries the positive sign ($V = +0.220$) and telecom the negative ($V = -0.173$), with retail near zero ($V = -0.012$). StrictWrite again improves on NoGuard in point estimate, 0.192 task success against 0.157. The 8,340-episode 7B run therefore changes the backbone without changing the qualitative regime.

N Cross-Benchmark Transfer

To check whether the negative contrast is specific to τ^2-bench, we run the audit on a second benchmark family. We use the AgentDojo workspace suite's benign user tasks (v1.2.2), comprising 40 tasks over email, calendar and cloud-drive tools, with no prompt-injection attacks. The agent is Qwen2.5-14B at $B = 10$, three seeds, with the six AgentDojo-compatible arms (NoGuard, TypeBlock and its zero-overhead variant, ReplanOnce, BestOf3ActionVerifier and Oracle). NecessityGate, ConfirmBeforeMutate and StrictWrite require τ^2-bench machinery that AgentDojo does not provide. TypeBlock blocks a curated set of ten workspace mutation tools that send or delete email, create, cancel or reschedule calendar events or add participants, and create, delete, append to or share files. A task is gold-irreversible when its reference solution calls one of them. The official per-task utility functions score success.

Table N.1: Cross-benchmark transfer on the AgentDojo workspace suite at $B = 10$, Qwen2.5-14B, three seeds, six arms. TSR is task success over the 11 capable tasks whose gold trajectory uses a mutation tool (the NoGuard or oracle arm succeeds on some seed). TypeBlock executes zero mutations and so fails every one, giving a task-clustered bootstrap guard value $V = -0.727\,[-0.909, -0.545]$. The positive side is structurally sparse here because mutation-free workspace goals are query tasks scored on the text answer. The transfer claim therefore concerns only the negative contrast.

Arm	Capable-mutation TSR (11 tasks)
NoGuard	0.727
TypeBlock	0.000
TypeBlock-ZeroOvhd	0.000
ReplanOnce	0.758
BestOf3ActionVerifier	0.394
Oracle	0.758

Table N.1 shows the replication. Of the 40 tasks, 22 are gold-irreversible and 11 of those are capable (the NoGuard or oracle arm succeeds on some seed). On those 11 capable tasks whose gold trajectory uses a mutation tool, the unguarded agent succeeds at 0.727 while TypeBlock, which executes zero mutations, drops to 0.000, giving $V = -0.727$ with a task-clustered interval of $[-0.909, -0.545]$. BestOf3ActionVerifier reaches only 0.394 there, below NoGuard, and ReplanOnce and the oracle both reach 0.758, near NoGuard. The mutation-free tasks are query tasks scored on the text answer, so a spurious mutation need not break the goal as it does in τ^2-bench, and we make no claim that the positive contrast transfers. On these 11 selected tasks, AgentDojo therefore replicates the negative contrast.

O Reproducibility Details

The anonymized reviewer artifact contains offline analysis code, non-sensitive configuration, de-identified per-episode records and reference reports. Its commands recompute the main branch contrasts, occupancy estimates and validity counts from the released records. The accompanying documentation distinguishes runnable analyses from reference-only reports. For the audit, one offline analysis rebuilds the stratified table, the pooled row and the mean tokens per arm from the records without third-party packages. A second analysis uses NumPy and the same records to recompute the task-clustered intervals, break-even weights, cross-fitted choice and capability table. Comparison with the archived report finds zero differences on the released data. The artifact cannot recompute the live scorer replay or generate new episodes. These require the benchmark, served models and execution dependencies outside the offline package. Full conversation traces and the figure-rendering workflow are not included. The primary benchmark is a fixed revision of verified τ^2-bench, using its airline, retail and telecom pools. The transfer benchmark is the AgentDojo workspace suite (v1.2.2) (Debenedetti et al., 2024). The agent backbones are fixed Qwen2.5-Instruct 14B and 7B checkpoints (Qwen et al., 2025) under activation-aware weight quantization (AWQ) (Lin et al., 2024), four bits and group size 128. Table O.1 gives the two vLLM serving stacks (Kwon et al., 2023) used for the audit and the branch experiments. No estimate is computed as a difference across them.

Each experiment fits a single GPU, and Table O.1 records the relevant package versions. A per-episode budget meter charges guard checks, replans, confirmations, candidate rollouts, verifier calls, and executed tools to the arm, recording an over-budget arm at its last executed state. Each audit condition uses three seeds, and because the seeds of a task are not independent the resampling unit for inference is the task. From the released records the scripts reproduce the regime stratification, the guard value within each stratum, the per-arm budget and token means, the cross-fitted choice and the task-clustered intervals. The confirmatory analysis itself still needs the per-episode traces, which are too large to release, so the archive carries its output as well. The offline scripts can check that output, so it need not be taken on trust.

Table O.1: The two serving stacks used for the reported experiments. Both stacks use the same weights, benchmark revision and decoding settings. Server and framework versions and GPUs differ, so every comparison in this paper stays within the audit or within the branch experiments.

	audit half	branch half
vLLM	0.24.0	0.8.5.post1
PyTorch	2.11.0	2.6.0
transformers	5.12.1	4.51.3
numpy	2.3.5	2.2.6
Python	3.11.15	3.11.15
CUDA compiler	12.2	11.4
GPU	L20, 46 GB	A100-SXM4, 80 GB
actor and customer	temperature 0, 512 tokens	temperature 0, 512 tokens
budget and steps	$B = 10$, 30 proposals	$B = 10$, 30 proposals

P BRANCHING A LIVE BENCHMARK EPISODE

The instrument requires two continuations from one prefix. Figure 1(c) gives the construction. The benchmark offers no rollback and its environment is stateful. A background thread manages the episode's conversation with a simulated user. Branching therefore requires a live copy rather than replay.

Copied state. The environment is deep-copied at the decision point. A plain copy fails, because the live episode holds threading primitives that cannot be serialized. The copy therefore substitutes fresh primitives for the five that the running episode owns. These are the environment lock, the simulation-done event, the orchestrator thread handle, and the agent's own lock and turn-finished event. All other state, including the database and complete conversation, is copied unchanged.

Resumption point. A clone must not re-run the step at which it was captured. The resume path skips observation and re-enters the loop where the episode awaited an action. The branch therefore begins at exactly the parent's decision point. It reproduces the second half of the benchmark's own turn method. The branch contract asserts the sequence that clears the turn event, copies the message history into the current observation, waits for the resumed turn and appends its reply. A benchmark upgrade that moves any of these operations raises an error rather than producing branches that start half a step early.

The same contract specifies which objects are copied and which are replaced. Everything reachable from the episode is deep-copied, including the environment, database, complete message history and evaluator. Only the five handles above are replaced through the copy's memo, without patching the benchmark source. The copy is taken under the episode's lock, so no step can be in flight while it runs. The budget meter is forked beside the episode rather than copied with it. Each branch inherits the parent's interaction and token use at the decision point, then maintains its own counter. Neither draws on the other's budget, and both inherit the same remaining allowance. Deterministic tests check this independent accounting for four policies and selected scripted actions. Two additional retail cases check a customer continuation and a state-writing proposal through the live benchmark adapter. These bounded tests do not prove action-for-action replay under live model generation.

Branch closure. A clone resumes a thread that then blocks waiting for an action from a caller that will never send one. Those daemon threads persist without an immediate failure and accumulate. Closing sends the benchmark's own end-of-conversation token, which ends the simulation normally. The orchestrator marks the run terminated, finishes its loop, and the thread exits. Closure occurs only after the branch outcome is recorded, because closing appends a message.

Without closure, parked threads accumulate until the host's per-user thread limit stops new episodes. Tests that count live branch threads assert that every branch is closed.

The contract, as pseudocode. Algorithms P.1 to P.3 state the instrument at the level a reimplementation needs. The implementation checks each precondition and raises an error if one fails.

Algorithm P.1 Branch a live episode at a decision point

Require: live episode E parked at an agent decision point, budget meter M
Ensure: branch E' sharing no mutable state with E, meter M'

check E has not terminated, it is the agent's turn, the pending message is addressed to the agent, and the agent-turn source markers the resume path depends on are present
acquire E's episode lock, failing if a step is in flight
$E' \leftarrow$ deep copy of E, substituting fresh handles for the episode lock, the simulation-done event (same set state), the orchestrator thread (none), the agent lock and the agent's turn-finished event (same set state)
release the lock {E is untouched and can still be stepped}
in E', replace the orchestrator's initialization with a no-op {the task's initial actions were applied in the parent}
in E', install a one-shot agent turn that skips the observation phase, waits for the supplied action and appends it {the copied history already holds the observation}
start E''s orchestrator thread and wait until it re-enters the agent turn, failing on timeout or on termination
$M' \leftarrow$ a new meter carrying M's interaction and token usage and ledger {neither can charge the other}
return (E', M')

Algorithm P.2 Continue an episode under a guard policy, using the same loop for branches and whole episodes

Require: episode E, meter M, history h, policy π for later proposals, optional pending proposal a with its policy π_1, steps left n, optional stop count j

$s \leftarrow 0$ {mutating proposals seen}
for $i = 0$ **to** $n - 1$ **do**
  if $i = 0$ and a is given **then**
    $p \leftarrow a$, $\rho \leftarrow \pi_1$ {already proposed and charged in the parent}
  else
    stop if M is exhausted or E is done
    $p \leftarrow$ actor(h), record its tokens on M without a slot, $\rho \leftarrow \pi$
  end if
  if p has no tool call **then**
    stop if p has no text, else append it to h, append the simulated customer's reply, stopping if there is none, and **continue**
  end if
  $c \leftarrow$ the first tool call of p, $mut \leftarrow c$ is in the irreversible taxonomy
  if mut **then**
    $s \leftarrow s + 1$, and **return** (p, i) as a branch point if $s = j$
  end if
  if mut and ρ is a refusal **then**
    $x \leftarrow$ nothing, and append the harness's block notice to h if ρ is the noticed refusal
  else if mut and ρ is the engine **then**
    $x \leftarrow$ the engine's decision, its reviews charged to M, stopping if M runs out
  else
    $x \leftarrow c$
  end if
  continue if x is nothing {a refusal charges nothing}
  charge one interaction to M, stopping if it runs out, then execute x in E and append the result to h
end for

Deterministic tests check three instrument properties. In those fixtures, a prefix run to a mutating proposal and then continued under a policy reproduces the uninterrupted episode under that policy, action for action. Stopping at the first of n mutating proposals reproduces the earlier stopping rule field by field. A closed branch leaves no live thread behind.

Algorithm P.3 Run the occupancy walks of E8 over the points a guarded policy visits

Require: task and seed, guard policy π_g, base policy π_0, the points to measure S: every point, or, when a counting pass returned $K_c > 4$, the positions $\{1, \lceil K_c/3 \rceil, \lceil 2K_c/3 \rceil, K_c\}$ together with every $t \geq K_c - 2$, where $K_c \leq 4$ and $K_c = 0$ both measure every point

start episode E with a fresh meter M and n steps, and run Algorithm P.2 under π_g with $j = 1$
$t \leftarrow 0$
while the loop returned a branch point (p, i) **do**
  $t \leftarrow t + 1$, $n \leftarrow n - i$, and **break** if $n \leq 0$ {this point counts toward K even though it is not measured}
  if $t \in S$ **then**
    for arm $(\pi_1, \pi) \in \{$treatment (π_g, π_0), control (π_0, π_0), A/A twin $(\pi_0, \pi_0)\}$ **do**
      $(E', M') \leftarrow$ Algorithm P.1 on (E, M), then Algorithm P.2 on E' with pending p under π_1, later π, n steps
      record the reward of E' by the benchmark's own scorer, then close E' with the benchmark's end-of-conversation token
    end for
  end if
  continue E by Algorithm P.2 with pending p under π_g, later π_g, n steps, $j = 2$ {the parent moves only after its clones are taken}
end while
$K \leftarrow t$, $W \leftarrow$ the reward of E, and $\Delta_t \leftarrow R(\text{treatment}_t) - R(\text{control}_t)$ at each measured t

Q Separate Interaction and Intervention Budgets

The audit half charges every operation to one per-episode interaction budget. The branch half keeps a second, separate budget for interventions.

The budgets are separate because exhaustion should have different consequences. Running out of interaction budget ends the episode. Running out of intervention budget should not. The guard may no longer override, while the agent carries on working. Combining the budgets makes that state unreachable, although a controller allocating a scarce override budget would have to represent it. The two exhaustion signals are therefore distinct types, and the intervention one is deliberately not a subclass of the other, so a handler for one cannot silently catch the other.

Reviewing is charged as an interaction. Only a block or a replacement is charged as an intervention. A review that ends in allowing the proposed action costs the interaction and nothing more, so the engine's reviews and its overrides are counted separately.

No experiment reported in this paper sets a cap on interventions. The second budget exists in the harness and is exercised by its tests, but every run here leaves it unbounded. The state in which the guard may no longer override is therefore never entered, and the intervention budget is not part of the decision state in Equation (1). The branch fork carries interaction and token usage but not an intervention cap, so the branch implementation supports only this uncapped setting. In this paper the separation matters for accounting. A review that allows costs an interaction and is not an intervention, so the engine's firing count is its overrides and not its reviews.

The reviewed engine, as configured. The engine ran with at most two replacement candidates, 512 tokens and temperature 0 per call. Its history held at most 40 messages truncated to 4,000 characters, replacement was allowed, and a review whose verdict line could not be parsed executed the proposed action. The fixed review prompt asks for an execute-or-intervene verdict followed by a one-sentence reason, and the parser reads the first verdict it finds. On intervention, the replan prompt asks for one better tool call or none, and the engine blocks when no replacement is returned. The noticed arms tell the actor that the proposed action was blocked and not executed, then ask it to choose another step or explain the limitation. ReflectRepair appends its hint to that notice.

Table R.1: Values of Figure 2(a) to (d), with task-clustered 95% intervals. Per-point estimates use 270 retail branch points from 103 tasks and 115 airline branch points from 41 tasks. The middle column uses 342 retail and 150 airline task-seed units, assigning zero when no mutating proposal is reached, and is post hoc. Retail occupancy sums use 325, 342 and 341 units, respectively. Bold entries are pre-registered primary first-point tests. The airline noticed-refusal occupancy cell uses 123 units from 48 tasks after technical recovery. The technical repeat uses 150 units from the same 50 tasks, and its interval lies within the registered ± 0.10 band (Section 6.3). Dashes mark quantities without a comparable branch-record estimate, and cells marked not measured were never run. †Deterministic serving, effect from separately started whole episodes, on 50 units except the every-visit retail row, which uses the 25 informative tasks (Appendices Z.4, Z.5 and Z.6). The 7B rows use Qwen2.5-7B as actor and customer on 44 branch points and 50 units.

	$D - \Delta_1$, per point	$k\,\Delta_1 - D$, per unit	$\sum_t \Delta_t - D$, per unit
retail, silent	$\mathbf{-0.093}\,[-0.160, -0.026]$	$+0.149\,[+0.032, +0.330]$	$+0.023\,[-0.040, +0.078]$
retail, notice	$-0.037\,[-0.085, +0.004]$	$+0.012\,[-0.067, +0.094]$	$+0.006\,[-0.018, +0.029]$
retail, engine	$-0.004\,[-0.011, +0.000]$	$+0.006\,[+0.000, +0.015]$	$+0.000\,[-0.012, +0.012]$
retail, silent, every visit	–	–	$0.000\,[0.000, 0.000]^{\dagger}$
airline, silent	$\mathbf{+0.261}\,[+0.133, +0.398]$	$-0.007\,[-0.273, +0.420]$	not measured
airline, notice	$+0.174\,[+0.072, +0.287]$	$-0.080\,[-0.200, +0.027]$	$-0.041\,[-0.125, +0.041]$
airline, notice (repeat)	–	–	$+0.007\,[-0.060, +0.080]$
airline, engine	$+0.052\,[+0.000, +0.116]$	$-0.040\,[-0.093, +0.000]$	$0.000\,[0.000, 0.000]^{\dagger}$
7B airline, silent	$\mathbf{+0.227}\,[+0.114, +0.364]$	$-0.200\,[-0.320, -0.100]$	not measured
7B airline, notice	$+0.182\,[+0.068, +0.295]$	$+0.060\,[-0.280, +0.480]$	$0.000\,[0.000, 0.000]^{\dagger}$
7B airline, engine	$0.000\,[0.000, 0.000]$	$+0.040\,[0.000, +0.120]$	$0.000\,[0.000, 0.000]^{\dagger}$

R Estimators and the Pre-Registration Record

R.1 Estimator Map

Table R.2 lists every reported branch-experiment quantity, its unit-level form, source records and averaging support. Studies retain their registered labels, E1 to E8 and E10 for the branch experiments and P0 to P4 for later follow-up measurements. Table R.4 lists their hypotheses. Here $q_i[a]$ is the terminal reward of arm a on unit i, P_{once} applies the guard at the first mutating proposal only and P_{all} at every one. The paper reports three count-scaled quantities that are not interchangeable. The branch form multiplies the first-point value by the firing count of the always arm within one branch record. The walk form multiplies a first value re-measured on an occupancy walk by that walk's visit count, which for the engine includes reviews that allow. The pooled product multiplies two averages. All intervals are task-clustered bootstraps with 10,000 resamples. Table R.1 lists the values plotted in Figure 2.

Measurement cost. Table R.3 reports each occupancy campaign's additional clones and actor tokens per task-seed unit beyond one guarded episode, plus the unit's total wall time. Token counts subtract the parent's usage at capture. For campaigns that measured every visit, the last column gives the clones the fixed four-point schedule of Appendix W would have taken on the same walks. A schedule also needs a counting episode, which for silent refusal in E8 cost about 131 thousand actor tokens per unit. Full coverage costs about as much as a schedule when the guard is visited two or three times per episode. For silent refusal, with 22 visits, it requires 3.8 times as many clones. Wall times depend on the server and its load and compare units within a campaign only. The simulated customer's tokens are not metered.

R.2 Pre-Registration Commitments

Each study's decision rule, thresholds and possible outcomes were specified before data collection. Its analysis code was fixed before its data were read. Table R.4 lists each study's question, rule, status and outcome. The E10 technical repeat was specified after the recovered E10 result had been inspected (Appendix W.1), and E6 was registered after the truncation defect of Correction 1 in

Table R.2: The estimators behind the branch-experiment results. Units are task-seed pairs. A unit that reaches no mutating proposal contributes zero wherever the support column says all units. Failed clones are excluded and never set to zero.

Quantity	Unit-level form	Records and support	Reported in
First-point value $\Delta_{1,i}$	$q_i[P_{\text{once}}] - q_i[\pi_0]$ at the first mutating proposal	branch records (E6, E7, 7B), branch points or all units	Table R.1, Figure 2(a)
Within-prefix effect D_i	$q_i[P_{\text{all}}] - q_i[\pi_0]$ from the same prefix	branch records, as above	Table R.1
Registered gap	$D_i - \Delta_{1,i}$	branch records, branch points	Table R.1, first column
Branch count-scaled error	$k_i\Delta_{1,i} - D_i$, k_i the firings of P_{all}, overrides for the engine	branch records, all units	Figure 2(b), Table R.1, second column
Pooled product	$\bar{k}\,\bar{\Delta}_1 - \bar{D}$	branch records, branch points	Appendix U
Walk count-scaled error P_i	$K_i\Delta_{1,i} - D_i$, with K_i visits and $\Delta_{1,i}$ both from the walk	occupancy walks, units with a measured first visit	Section 6.3, Table W.1
Occupancy sum S_i, or its scheduled estimate	$\sum_{t \le K_i} \Delta_{t,i}$ with every visit measured. The scheduled estimate is $\Delta_{1,i} + (K_i - 2)\bar{\Delta}_{\text{int},i} + \Delta_{K_i,i}$ under the fixed schedule	occupancy walks, units whose required visits are measured	Table R.1, third column
Paired policy effect	E8 and E10 take the unit's D_i from E6 or E7, the deterministic checks guarded minus unguarded whole episodes	policy campaigns	Section 6.3
Occupancy residual	mean of $S_i - D_i$	walks paired with the effect	Table R.1, third column
Walk effect W_i	walk reward minus the first-visit execute clone's reward	occupancy walks	Appendix Z.3
Sensitivity range	each missing unit's S_i set to -1 and $+1$	units without the required visits	Appendices W and W.1

Table R.3: Measurement cost per task-seed unit. Visits is the mean number of guarded-policy visits K. Tokens counts actor tokens inside the clones, in thousands. The schedule column is the clone count a four-point schedule would take on the same walks, shown only for campaigns that measured every visit. The E8 silent campaign used the schedule itself.

Campaign	Coverage	Units	Visits	Clones	Tokens (k)	Minutes	Schedule clones
E8 retail, silent	schedule	342	17.7	16.6	343.6	10.5	–
E8 retail, noticed	every visit	342	2.0	6.0	165.4	8.6	5.5
E8 retail, engine	every visit	342	1.6	4.9	113.9	5.3	4.9
E10 airline, noticed, repeat	every visit	150	1.8	5.5	130.1	3.2	5.3
P1 airline, noticed	every visit	50	1.8	5.5	142.3	5.8	5.4
P2 airline, engine	every visit	50	1.8	5.4	126.8	5.5	5.4
P3 retail, silent	every visit	25	22.0	65.9	1,110.7	31.5	17.3
P4 airline 7B, noticed	every visit	50	3.3	9.9	458.5	8.6	8.4
P4 airline 7B, engine	every visit	50	2.7	8.2	317.8	6.8	8.2

Appendix Y was found. Amendments that changed a rule are described with the study they concern. Amendments that only moved servers between machines are not listed.

S Excluded Runs of the First, Truncated Instrument

E1 to E3 ran on an implementation whose branches ended at the agent's first message to the simulated customer rather than at the end of the episode (Correction 1 in Appendix Y). Their one-step values therefore do not estimate Equation (1), and no claim in this paper rests on them. E1 asked

Table R.4: Registered question, decision rule, status and outcome of each study. A primary test decides the registered question, a secondary test uses the same rule, a reported quantity is descriptive, and a post hoc analysis was specified after the data. E1 to E3 ran on the truncated first instrument and are excluded from every claim. E8's three arms are listed separately because their outcomes differ. The E10 repeat, marked later, was specified after the recovered E10 result had been inspected and before its own data were collected.

	question	rule	status	outcome
E1	the engine harms	negative mass ≥ 0.05	primary	fail, 0.0213
E2	blanket blocking harms	negative mass, then sign test	primary	outcome (c)
E3	the estimator's noise	median pair range vs 0.5	primary	resolves, 0.0
E4	$k\overline{\Delta}$ predicts D	in interval	primary	no
E5	the floor is the actor's	$C_{32} \geq 42$ or ≤ 31	primary	pool, $C_{32} = 21$
E6	one refusal stands in	gap excludes zero	primary (silent)	no, -0.093
	with a notice or a review	same rule	secondary	-0.037, -0.004
	$\bar{k}\,\bar{\Delta}_1 - \bar{D}$, silent, noticed	none	reported	$+0.175$, -0.042
E7	E6's rejection on airline	gap excludes zero	primary (silent)	replicated, $+0.261$
	airline, notice or review	same rule	reported	$+0.174$, $+0.052$
	the noticed gap, pooled	interval and width	primary	mis-specified
E8	visited values sum to D	R_{occ}, then H	primary (silent)	unresolved, 0.118
	the same, noticed refusal	same rule	secondary	compatible, 0.047
	the same, engine	same rule	secondary	compatible, 0.023
	P, silent, noticed	none	reported	$+0.070$, -0.020
	the within-run split	none	post hoc	in Appendix W
E10	airline occupancy replication	validity, then tolerance	primary	unresolved
	repeat on same tasks	same rule	later	within, P unresolved
P0	copy or serving	re-send agreement	primary	deterministic agrees
P1	same-regime sum equals D	± 0.10 band	primary	within, -0.020
P2	the same, reviewed engine	same rule	primary	within, 0.000
P3	silent sum, every visit	same rule	primary	within, 0.000
	fixed schedule, same walks	differs from full sum	secondary	differs, unresolved
P4	silent gap, second actor	gap excludes zero	primary	replicated, $+0.227$
	noticed and engine sums	± 0.10 band	secondary	within, 0.000
	order of four policies	against whole episodes	secondary	Δ_1, $k\Delta_1$ reverse it

Table T.1: E4 over four arms of 342 episodes each. Four silent-refusal episodes failed, 1.17% against a 5% ceiling, and are excluded from every aggregate. Opportunities are mutating proposals seen, and firings are times the policy acted on one.

policy	completed	mean reward	opportunities	firings k
no guard	342	0.152	1.77	0.00
silent refusal	338	0.089	18.51	18.51
noticed refusal	342	0.088	2.08	2.08
reviewed engine	342	0.146	1.65	0.49

whether the reviewed engine's interventions cost success, E2 whether one blocked action costs success, and E3 how much repeated continuations vary. With the corrected loop and a larger pool, E6 replaces E2, and its engine arm answers E1's question (Appendix U). Later experiments use execute/execute pairs to measure the re-run variability addressed by E3. The records of E2 and E3 are included in the code supplement.

T Experiment 4: Whole-Episode Policies

E4 runs the same four policies as full-horizon whole episodes with the same pool, budget and actor. Its policy effects serve as the between-episode comparator for the retail branch experiment (Appendix U).

Table U.1: E6 estimates, with task-clustered 95% intervals from 10,000 resamples. Per point denotes averages over the 270 branch points and per unit averages over all 342 units, with zeros where no branch point was reached. The per-unit policy effect is on the scale of E4's per-episode one, and the last row is the pre-registered instrument check against it.

	silent refusal	noticed refusal	reviewed engine
one-step Δ, per point	$+0.0037$ $[+0.0000, +0.0116]$	-0.0519 $[-0.1123, +0.0073]$	$+0.0037$ $[+0.0000, +0.0116]$
one-step Δ, per unit	$+0.0029$ $[+0.0000, +0.0088]$	-0.0409 $[-0.0906, +0.0059]$	$+0.0029$ $[+0.0000, +0.0088]$
policy effect D, per point	-0.0889 $[-0.1579, -0.0223]$	-0.0889 $[-0.1579, -0.0223]$	$+0.0000$ $[-0.0110, +0.0111]$
policy effect D, per unit	-0.0702 $[-0.1257, -0.0175]$	-0.0702 $[-0.1257, -0.0175]$	$+0.0000$ $[-0.0088, +0.0088]$
gap $D - \Delta$, per point	-0.0926 $[-0.1602, -0.0263]$	-0.0370 $[-0.0852, +0.0037]$	-0.0037 $[-0.0114, +0.0000]$
$D_{\text{in}} - D_{\text{E4}}$, per task	-0.0080 $[-0.0350, +0.0199]$	-0.0059 $[-0.0322, +0.0205]$	$+0.0059$ $[-0.0146, +0.0292]$

The paired bootstrap resamples the 114 tasks 10,000 times, keeping each task's three seeds together. Silent refusal changes success by $-0.062\,[-0.118, -0.009]$ against no guard, and the reviewed engine by $-0.006\,[-0.029, +0.015]$. Silent refusal fires 18.51 times per episode against 2.08 for noticed refusal, and their mean rewards are 0.089 and 0.088.

Archived composition test. E4's registered composition test multiplied the firing count by E2's one-step values, which come from the truncated loop (Appendix S), so it supports no claim. Silent refusal's 18.51 firings per episode, against 1.77 mutating opportunities in an unguarded episode, are consistent with repeated proposals. The corrected test uses one prefix and one loop (Appendix U).

U Experiment 6: One Refusal or One Policy, From the Same Prefix

E6 replaces E2. It was pre-registered after the defect of Correction 1 was found, and its analysis code was fixed before any of its data existed. Every continuation uses the whole-episode loop, so the simulated customer answers the agent inside a branch as it would in an unbranched run. Deterministic tests check action-for-action agreement for four policies on selected scripted cases. They do not establish exact replay under live model generation.

Scale and validity. Across 114 retail tasks under outer seeds 0, 1 and 2, all 342 units completed. Of these, 270 reached a mutating proposal and were branched into seven arms. None of those branches failed. The 72 units without a mutating proposal reached no guard decision and receive zero policy effect in every per-unit average. There were 51 informative branch points across 25 tasks, exceeding the pre-registered floor of 25. One continuation per arm was run with greedy actor decoding and a zero-temperature simulated user. The outer seeds are repeated runs within tasks, and the bootstrap resamples tasks. Appendices W and Z report re-run variability measured with paired execute clones.

Arm-level outcomes. Against executing the proposal, a single silent refusal changes the outcome at one branch point, raising it. A single noticed refusal changes the outcome at 30, increasing it at eight and decreasing it at 22. Refusing always changes it at 38, seven up and 31 down. It does so identically with and without the notice at all 270 points, because neither variant executes a mutating call after the branch point and the prefix has executed none. At the seven points where repeated refusal helps, the execute branch failed and the refusal branch succeeded, matching the direction in the audit's positive stratum (Section 3). The engine's one-step arm changes the outcome at one point and its policy arm at two, one each way.

The silent single refusal took exactly one more step than the executed proposal at 229 of the 270 points, consistent with delay. The records carry step counts rather than the next proposed action, so they cannot establish a repeated call. Continuations ended naturally in 260 of 270 unguarded arms and 258 silent one-step arms. Refusing always without the notice ran to the step limit in 266, as in E4, and with the notice the conversation ended naturally in 269. The engine's arms ran out of budget in 73 one-step and 140 policy continuations, because reviewing is charged to the same budget the task uses.

The composition product. The pre-registration lists the product $k\,\Delta$, with the uncertainty of Δ carried forward, as a descriptive quantity. Its interval was computed after the data by resampling k, Δ and D together over tasks. For the silent refusal the product is $+0.0859\,[+0.0000, +0.2699]$ against a policy effect of -0.0889, and the product minus the effect is $+0.1747\,[+0.0436, +0.3847]$. With the notice the product is $-0.1312\,[-0.2813, +0.0184]$ and the difference is $-0.0423\,[-0.1503, +0.0705]$. For the engine both are zero to within 0.012. The two comparisons answer different questions. The first-point gap compares refusing always with refusing once, and for the noticed refusal its interval, at width 0.0889, meets the registered width criterion, which is not an equivalence margin. The product asks whether the policy's cost is its refusal count times the one-step value, and at width 0.2208 it does not resolve.

Text-channel and unmatched-seed checks. Independent recomputation from the raw records with separate code found no discrepancy in any estimate above. Two further checks concern text-channel calls and unmatched seeds. First, a reply with no parsed tool call goes to the simulated customer. If the whole reply is a tool-call string, the benchmark executes it as a tool call, which would bypass the guard. A call arriving that way leaves a text turn that no simulator reply answers. Across the 1,022 completed E4 episodes under the three non-reviewing policies, every such unanswered turn falls at the benchmark's own 30-step cap, where the last turn is never answered, and none falls below it. Second, four E4 silent-refusal episodes failed. Because the instrument check aligns by task, it averages over slightly different seeds in the two comparisons. On the 338 matched pairs the blocking difference is $-0.0059\,[-0.0325, +0.0208]$, and the conclusion is unchanged.

A silent refusal against a noticed one. At the same points, the silent single refusal minus the noticed one is $+0.0556\,[-0.0038, +0.1174]$. At this resolution the separate effect of the notice is not distinguished from zero.

V Experiment 7: A Second Domain

E7 repeats E6 on all 50 airline tasks under three seeds, 150 units. It also runs E4's four policies as whole episodes on the same pairs, 600 episodes, so the instrument can be checked there too. Its pre-registration explicitly states that it followed knowledge of E6's result. Before the full run, one amendment changed the replication rule from one-sided, below zero, to two-sided, matching E6's rule. The only airline observation available then was one excluded test unit with a positive gap. The analysis code was fixed before the full-run data were read.

Scale and validity. All 150 units completed, and 115 reached a mutating proposal. No branch failed. The 54 informative branch points spanned 20 tasks, exceeding the floor of 12. All 600 episodes completed. Mean reward between episodes is 0.26 unguarded, 0.46 refusing with or without the notice and 0.313 under the reviewed engine.

Arm-level outcomes. The silent single refusal took exactly one more step than the executed proposal at 98 of 115 points and changed the outcome at one. Refusing always changed it at 31, all 31 for the better, identically with and without the notice at all 115 points. The single noticed refusal changed it at 19, 15 up and 4 down. The engine's policy arm changed it at 13, 10 up and 3 down. Between episodes, the estimated effect of refusing is $+0.2000\,[+0.0867, +0.3200]$ with or without the notice and the engine $+0.0533\,[+0.0000, +0.1200]$, and the unguarded branch arm matches the unguarded episode at 139 of 150 pairs.

Limitations of the pooled test. The rule for the noticed refusal pooled its first-point gap over retail and airline with a bootstrap stratified by domain. It returns $+0.0260\,[-0.0184, +0.0709]$, which the

Table V.1: E7 estimates on airline, task-clustered 95% intervals, 10,000 resamples, seed 20260913. Per point denotes averages over the 115 branch points and per unit averages over all 150 units. The last row is the instrument check against the airline whole-episode run.

	silent refusal	noticed refusal	reviewed engine
one-step Δ, per point	$+0.0087$ $[+0.0000, +0.0270]$	$+0.0956$ $[-0.0089, +0.2035]$	$+0.0087$ $[-0.0696, +0.0840]$
one-step Δ, per unit	$+0.0067$ $[+0.0000, +0.0200]$	$+0.0733$ $[-0.0067, +0.1600]$	$+0.0067$ $[-0.0533, +0.0667]$
policy effect D, per point	$+0.2696$ $[+0.1368, +0.4118]$	$+0.2696$ $[+0.1368, +0.4118]$	$+0.0609$ $[-0.0348, +0.1624]$
policy effect D, per unit	$+0.2067$ $[+0.1000, +0.3200]$	$+0.2067$ $[+0.1000, +0.3200]$	$+0.0467$ $[-0.0267, +0.1267]$
gap $D - \Delta$, per point	$+0.2609$ $[+0.1327, +0.3982]$	$+0.1739$ $[+0.0721, +0.2870]$	$+0.0522$ $[+0.0000, +0.1157]$
$D_{\text{in}} - D_{\text{E4}}$, per task	$+0.0067$ $[-0.0467, +0.0667]$	$+0.0067$ $[-0.0467, +0.0667]$	-0.0067 $[-0.0467, +0.0267]$

rule classifies as agreement, although the domain gaps have opposite signs. The per-domain gaps are -0.0370 in retail and $+0.1739$ in airline. Because the policy's effect has opposite signs in the two domains, the pooled policy effect is only $+0.0182$ and the signed gaps cancel. The same pooling gives silent refusal $+0.0130\,[-0.0485, +0.0746]$. This interval includes zero although the gap excludes zero in each domain separately, so the per-domain gaps above are the informative quantities.

The composition product. For the silent refusal the product is $+0.2318\,[+0.0000, +0.7266]$ against a policy effect of $+0.2696$. The product and the policy effect are both positive here, but the product's interval is too wide to show whether it predicts the effect. For the noticed refusal it is $+0.2279\,[-0.0207, +0.4779]$. The first-point gap, not the product, tests whether one intervention's value differs from the repeated policy's effect.

The text channel. In the 450 airline episodes under the three non-reviewing policies, 48 text turns went unanswered by the simulator. All fall at the benchmark's 30-step cap and none below it, so no reply below the cap was executed as a tool call.

W Experiment 8: The Identity Along the Guarded Policy's Own Visits

Experiments 6 and 7 test a substitution. They ask whether the one-step value where the unguarded policy first offers the guard a decision stands in for the policy effect. The identity of Section 4 sums one-step values over the points the guarded policy itself visits, weighted by how often it visits them, and E8 measures that sum. It asks two questions in order. The first is what the residual of that identity measures in this harness, where reviewing costs budget, the customer answers inside the loop and the state includes the interaction budget and a step limit. The second is which observable feature of a decision point says when a first-point estimate transfers, and the candidate named in advance was the remaining horizon. E8's rules, estimators, validity thresholds and possible outcomes were fixed before any E8 code or data existed, and each later amendment was made before the data it could affect.

Table W.1 separates the per-point first value from the per-unit product and occupancy residual. The first-point policy contrasts in Table R.1 condition on 270 retail or 115 airline branch points. They are from E6 or E7, not from the occupancy walks. For R_{occ} and P, D is re-paired to the exact retained unit support in each row. Missing clone values reduce that support. A policy that never fires contributes zero by definition. The numbers of successful clone attempts are 5,673/5,673, 2,064/2,064 and 1,671/1,674 for the retail rows, then 688/750 and 819/819 for the airline rows. On the technical repeat's full support, the E7 policy effect is $+0.20667\,[+0.10000, +0.32000]$, whereas

Table W.1: Support for the occupancy quantities, as task-seed units / independent tasks. Δ_1 conditions on a measured first visited point. $P = K\Delta_1 - D$ and $R_{\text{occ}} = K\overline{\Delta}_g - D$, with K visits and $\overline{\Delta}_g$ the mean measured one-step value, include units where the guard never fires, with zero contribution, and D is re-paired on each quantity's retained support. Failed clone outcomes are never set to zero. Under the every-point plan, a missing interior value is replaced by the mean of the surviving interior values. The retail policies share one policy-effect campaign, and both airline production attempts share another and the same 50 tasks. The technical repeat was run after the recovered result was known. Deterministic rows take D from separately started whole episodes, and the 7B rows use Qwen2.5-7B as actor and customer.

Domain	Guard policy, walk	Units	Δ_1	P	R_{occ}	Clones completed
retail	silent refusal	342/114	263/99	342/114	325/114	5,673/5,673
retail	noticed refusal	342/114	272/102	342/114	342/114	2,064/2,064
retail	reviewed engine	342/114	263/99	341/114	341/114	1,671/1,674
airline	noticed, recovered attempt	150/50	84/36	130/49	123/48	688/750
airline	noticed, technical repeat	150/50	121/46	150/50	150/50	819/819
airline	engine, deterministic	50/50	37/37	50/50	50/50	270/270
airline, 7B	noticed, deterministic	50/50	43/43	50/50	50/50	495/495
airline, 7B	engine, deterministic	50/50	43/43	50/50	50/50	408/408

the recovered E10 residual uses only 123 units from 48 tasks. Neither result should be substituted for a point-conditioned branch contrast.

Design. The study uses E6's 342 retail units across 114 tasks under outer seeds 0, 1 and 2. The actor, simulated customer, budget, step limit, irreversibility trigger and scorer are unchanged, so each retained E8 unit pairs with its E6 unit. For each unit the guarded policy runs from the start of the episode under its own decisions, and its mutating proposals are numbered $t = 1, \ldots, K$. The parent follows this trajectory once and is cloned at each measured point before continuing. A measured point yields three clones, each inheriting the parent's remaining budget and remaining steps. The treatment clone applies the guard at t and the base policy after it, and the control clone executes the proposal at t and runs the base policy after it. A second identical control forms an A/A pair with the first. Their difference measures re-run variability without intervention. The one-step value at the point is $\Delta_t = R(\text{treatment}) - R(\text{control})$. The noticed refusal and the reviewed engine are measured at every point they visit. The silent refusal made 23.18 refusals per branch point in E6. Because the next proposed action was not recorded, this count does not establish repeated calls. The silent policy is therefore measured at four positions fixed by a counting pass, $t = 1$, $\lceil K_c/3 \rceil$, $\lceil 2K_c/3 \rceil$ and K_c, where K_c is that pass's count rather than the walk's own. A walk can end before or after the counted position. It therefore also measures every point from two before K_c onward, capturing its own last point unless it ends more than two points early. Thus, measurement positions come from the counting pass and weights from the walk.

Estimators and rules. Every estimate is on the per-unit scale of E6's policy effect, with a zero for a unit whose policy visits no mutating point, and every interval is a task-clustered bootstrap with 10,000 resamples. Within a unit, $\overline{\Delta}_g$ is the mean one-step value over the visited points. For the silent refusal it is the stratified estimate $[\Delta_1 + (K-2)\,\frac{1}{2}(\Delta_{\lceil K/3 \rceil} + \Delta_{\lceil 2K/3 \rceil}) + \Delta_K]/K$, in which the two interior points represent the $K-2$ interior positions, and extra points near the end enter no average.

The released estimator specifies six further cases. The weights and both endpoints always use the walk's own K, whereas interior positions come from the counting pass. Walks longer or shorter than that pass are therefore weighted by their actual visits. A unit whose counted K is at most four, or zero, is treated as a unit measured at every point. With $K = 1$ the estimate is Δ_1 and with $K = 2$ it is $(\Delta_1 + \Delta_2)/2$. Under a plan that measures every point, a clone that failed at an interior point is replaced by the mean of the interior points that survived, weighted by K as above. A unit without a measured value at its first or last point is excluded from the occupancy residual rather than estimated. This accounts for the seventeen excluded units. A point reached with no continuation steps left is marked unmeasured and still counts toward K.

Table W.2: E8 estimates, with task-clustered 95% intervals from 10,000 resamples, seed 20260913. Each row averages over a different set of units and not over one common support. The columns are silent refusal, noticed refusal and the reviewed engine. K uses $342, 342, 342$ units. Δ_1 uses $263, 272, 263$ units with a measured first point. $\overline{\Delta}_g$ and R_{occ} use $325, 342, 341$ units with an occupancy estimate. P uses $342, 342, 341$ units and requires a first point or no visit. H uses $243, 174, 176$ units and requires both ends. Zeros enter where the policy visited no point, and D_{E6} is E6's policy effect on the same unit. On the units R_{occ} is computed over, P for the silent refusal is $+0.06462\,[+0.01216, +0.12037]$ rather than the $+0.07018$ this table reports over all 342 (Appendix W, and the support table released with the archive). The exploratory rows cannot change a registered outcome. For the noticed refusal they were computed after its registered result was known, and the last two rows were declared only for arms whose data had not been read.

	silent refusal	noticed refusal	reviewed engine
units complete, of 342	342	342	342
clones run, failed	5,673, 0	2,064, 0	1,674, 3
informative points	272	154	69
mean visited points K per unit	17.69883	2.0117	1.63158
units with a measured first point	263	272	263
units with an occupancy estimate	325	342	341
Δ_1, at the first visited point	0.00000 $[0.00000, 0.00000]$	-0.06250 $[-0.12057, -0.01103]$	-0.00380 $[-0.01181, 0.00000]$
$\overline{\Delta}_g$, per unit	-0.00239 $[-0.00595, +0.00023]$	-0.04769 $[-0.08470, -0.01597]$	-0.00196 $[-0.00880, +0.00292]$
identity residual R_{occ}	$+0.02308$ $[-0.03988, +0.07764]$	$+0.00585$ $[-0.01754, +0.02924]$	$+0.00000$ $[-0.01173, +0.01173]$
count-scaled error P	$+0.07018$ $[+0.01754, +0.12573]$	-0.02047 $[-0.11404, +0.06140]$	-0.00293 $[-0.01462, +0.00588]$
horizon contrast H	-0.02058 $[-0.07265, +0.03279]$	-0.01724 $[-0.11243, +0.07059]$	$+0.00568$ $[0.00000, +0.01796]$
A/A pairs that differ	4 of 1,891	3 of 688	0 of 555
Exploratory, declared in the third to fifth amendments			
D_{E8}, the walk's own effect	-0.04615 $[-0.09657, +0.00305]$	-0.06433 $[-0.12281, -0.01170]$	$+0.00000$ $[-0.00880, +0.00877]$
$R_{\mathrm{sched}} = K\overline{\Delta}_g - D_{\mathrm{E8}}$	$+0.00462$ $[-0.05452, +0.05183]$	$+0.00000$ $[-0.01170, +0.01170]$	$+0.00000$ $[0.00000, 0.00000]$
$R_{\mathrm{cross}} = D_{\mathrm{E8}} - D_{\mathrm{E6}}$	$+0.01846$ $[-0.00937, +0.04893]$	$+0.00585$ $[-0.01462, +0.02339]$	$+0.00000$ $[-0.01173, +0.01173]$
adjacent clone pairs that agree	1,187 of 1,187	682 of 688	557 of 557
R_{occ}, every measured point	$+0.02923$ $[-0.02322, +0.07855]$	–	
$\Delta_K - D_{\mathrm{E6}}$, last point alone	$+0.04923$ $[+0.01246, +0.08978]$	–	$+0.00000$ $[-0.01173, +0.01173]$

The identity residual R_{occ} is the mean over units of $K\,\overline{\Delta}_g - D_{\mathrm{E6}}$, paired with E6's own policy effect on the same unit. The count-scaled error P replaces the occupancy sum with $K\,\Delta_1$, and the horizon contrast is $H = \Delta_K - \Delta_1$. The rules are applied sequentially, starting with R_{occ}. If its interval excludes zero, the rule reports that the measurement violates a condition of the identity and does not evaluate H. Otherwise H decides for the silent refusal. An interval below zero would identify the remaining horizon as the feature. The registered rule labels an interval containing zero as falsifying the candidate, or as unable to distinguish when it is wider than 0.10. Without a negligible-effect margin, we report both outcomes as no detected contrast.

A run is valid with at least 300 complete units, a unit failure rate of at most 5%, a clone failure rate of at most 20% and at least 25 informative points. The A/A share is reported regardless of its value.

The noticed refusal. All 342 units completed, 2,064 clones ran and none failed, and 154 measured points were informative. The policy visits 2.01 mutating points per unit. Summed over those points, the one-step values are compatible with E6's policy effect within the measured precision. The resid-

ual is $+0.00585\,[-0.01754, +0.02924]$, an interval containing zero at a width of 0.047. That width is narrower than the agreement width E6 fixed in advance for a different comparison. That rule is a width threshold and not an equivalence test.

The count-scaled error, which substitutes the first visited point for all visits, is $-0.02047\,[-0.11404, +0.06140]$ and does not resolve at a width of 0.175. The horizon contrast over the 174 units that visit two or more points is $-0.01724\,[-0.11243, +0.07059]$, too wide at 0.183 to resolve the contrast. Three of the 688 A/A pairs differ, so a clone taken twice from one point almost always scores the same.

The silent refusal. All 342 units completed, 5,673 clones ran and none failed, and 272 measured points were informative. The policy visits 17.70 mutating points per unit, and its one-step value at the first of them is exactly zero in all 263 units that visit one. Seventeen units lack an occupancy estimate. Their walks ended three to ten points before the counted endpoint, beyond the two-point margin, leaving the last point unmeasured. On the other 325 the identity residual is $+0.02308\,[-0.03988, +0.07764]$. Of the 17, 14 have an E6 effect of zero and 3 an effect of -1. A post hoc sensitivity calculation that assumes each missing unit's occupancy sum lies in $[-1, 1]$, which holds for an exactly telescoping or expected sum but not automatically for a sampled one, places the point estimate over all 342 units in $[-0.01901, +0.08041]$. Measuring every visit on the 25 informative tasks resolves the sum there (Appendix Z.5).

The residual interval contains zero, so the rule does not reject the identity and proceeds to the horizon contrast. Its width of 0.118 is too large to interpret as agreement. The horizon contrast over the 243 units that visit two or more points is $-0.02058\,[-0.07265, +0.03279]$, which at a width of 0.105 the rule reports as unable to distinguish. The remaining horizon is therefore neither supported nor falsified as the feature determining when a first-point estimate transfers.

The count-scaled error is $+0.07018\,[+0.01754, +0.12573]$, E6's whole policy effect with its sign reversed, because the first point is worth nothing. Four of the 1,891 A/A pairs differ, and the interior spread averages 0.00429 over 233 pairs, one of them above 0.05. Forty-two silent-refusal units were complete before the completion run, as the third amendment below records.

The reviewed engine. After the completion that the sixth amendment describes, all 342 units completed. Three of 1,674 clones failed, and 69 points were informative. The engine visits 1.63 mutating points per unit. E6 had measured its policy effect at 0.000, so a near-zero occupancy residual was expected before the arm ran. The measured residual is $+0.00000\,[-0.01173, +0.01173]$ on 341 units. One unit carries no estimate, because a control clone at its first point timed out. The horizon contrast is $+0.00568\,[0.00000, +0.01796]$ and no A/A pair of 555 differs. In E6 the engine overrides 0.67 times per retail branch point.

Residual decomposition. This analysis was declared in the third amendment, before the silent refusal's completion run produced a record. For noticed refusal, the analysis followed knowledge of the registered result. It is therefore post hoc for that arm and cannot change a registered outcome. When adjacent continuations agree, the one-step values along one walk telescope, and each unit carries its own paired effect $D_{\mathrm{E8}} = W - V_1$, the walk's final reward minus the reward of its first control clone. The registered residual then splits exactly into $R_{\mathrm{sched}} = K\overline{\Delta}_g - D_{\mathrm{E8}}$, inside one campaign, and $R_{\mathrm{cross}} = D_{\mathrm{E8}} - D_{\mathrm{E6}}$, between two.

For the noticed refusal R_{sched} is $+0.00000\,[-0.01170, +0.01170]$, and a treatment clone scores what the next control clone scores, or at the last point what the walk scores, in 682 of 688 adjacent pairs. The observed point estimate of the registered residual therefore equals R_{cross}, $+0.00585\,[-0.01462, +0.02339]$. The two campaigns assign the same effect to 330 of the 342 units and different ones to 12. Within one run the scheduled residual is not detectably different from zero, and 6 of 688 adjacent pairs disagree. The difference from E6's effect lies between the two campaigns, and serving variation is one possible source (Appendix Z.2).

For silent refusal, the analysis was declared before its data were examined. On its 325 units R_{sched} is $+0.00462\,[-0.05452, +0.05183]$ and R_{cross} is $+0.01846\,[-0.00937, +0.04893]$, every one of the 1,187 adjacent clone pairs agrees, and the two campaigns assign different effects to 22 units. Of the 33 units with a nonzero walk effect, 13 show no nonzero one-step value at any measured point, so the schedule missed the visits that carry those effects. Of the other 20, 17 have one at

the last point, 3 at a margin point before it and 1 at a scheduled interior point, one unit in two of these places. Under the fifth amendment's breakdown by remaining steps, 17 of the 22 nonzero one-step values occur where a single step remains. Counting every measured point once and filling unmeasured positions from the scheduled interior, as the fourth amendment added, gives a residual of $+0.02923\,[-0.02322, +0.07855]$. The last one-step value alone, from the fifth amendment, falls short of E6's effect by $+0.04923\,[+0.01246, +0.08978]$. Most, but not all, measured nonzero silent-refusal values occur at the last opportunity. A single measurement at the horizon falls short of the policy effect. For the engine, declared before its data like the silent refusal, the within-campaign part is exactly zero and all 557 adjacent clone pairs agree.

Amendments. Six amendments were made, each before the data it could affect. The first replaced re-running the prefix once per measured point with the walk, because a prefix re-run one turn longer cannot reach the last point, which lies at the edge of the step budget. The second replaced a guarantee that the walk measures its own last point with the two-point margin, because the last point is known only once the walk has moved past it. The margin failed for the 17 silent units above. The third fixed the sampler for units whose counting pass found no mutating proposal. It also merged the silent refusal's completion run with the 42 units already complete, under a rule fixed in advance that required one record per unit and a merged total equal to E6's 342 units. The fourth corrected the analysis code to the registered stratified formula, before any value of the completion run was read, and declared the estimate that counts every measured point once. The fifth declared the last-point comparison and the count of nonzero values by steps left, for the silent refusal and the engine, whose data had not been read. The sixth re-ran, under the same merge rule, the engine units lost when its server stalled.

W.1 Experiment 10: Airline Occupancy After Technical Recovery

The airline replication measured noticed refusal at every visited mutating point on the same 150 task-seed units as Experiment 7. Its validity criteria required at least 132 complete units, at most 5% failed units, at most 20% failed clones and at least 12 informative points. These criteria preceded the primary comparison of the occupancy residual with a fixed ± 0.10 tolerance. The firing-count comparison was secondary and could be interpreted only if the primary residual lay within that tolerance.

The first pass completed 110 of 150 task-seed units, with 40 request timeouts, and a recovery pass fixed in advance re-ran those 40 units once and kept the 110 completed records. All recovery units completed without a further unit or clone failure. The merged run had 150 completed units, 62 failed clones among 750 attempts (8.27%), and 87 informative points, and met the numerical validity criteria. The original failed attempt is retained separately. Its automatically generated effect estimates had been viewed before the completion merge, contrary to the specified count-only review procedure. The recovered units were selected by failure status, and no threshold or estimator was changed.

The occupancy residual was $-0.04065\,[-0.12500, +0.04065]$, based on 123 task-seed units from 48 tasks. The remaining 27 units lacked measurement support because of retained clone timeouts in the original run. The diagnostic rule assigned 14 to missing first-point measurements, 12 to missing last-point measurements, and one to a missing scheduled interior measurement. Some units lacked both endpoints. Failures were not treated as zero effects. The residual therefore describes retained measurement support rather than the full task pool. A post hoc sensitivity calculation assumes that each missing unit's occupancy sum lies in $[-1, 1]$. The range holds for an exactly telescoping walk sum and an expected per-task policy effect with binary reward. A sampled sum of K paired differences can in principle reach $\pm K$, so the calculation is conditional, not a worst-case guarantee. Of the 27 missing units, 19 have an E7 effect of zero and 8 an effect of $+1$, so the point estimate over all 150 units lies in $[-0.26667, +0.09333]$. This range crosses the lower edge of the fixed ± 0.10 band, so the primary comparison is unresolved. The firing-count error was $-0.00769\,[-0.16129, +0.14394]$ on 130 units from 49 tasks. Its formal interpretation is withheld because the primary comparison did not establish agreement within tolerance. All intervals use 10,000 task-clustered bootstrap resamples with the original bootstrap seed.

On the primary comparison's 123 units, the registered secondary decomposition gives a within-campaign residual of $-0.03252\,[-0.08871, +0.02326]$ and a between-campaign component of

Table X.1: Distribution of E6's 51 informative points. All informative points occur on twenty-five of the pool's 114 tasks. Their rate within those tasks is more than four times the rate across all branch points. This concentration does not identify its cause.

		informative rate
all branch points, 103 tasks	270 points	0.189
tasks with any informative point	25 tasks	0.810

$-0.00813\,[-0.09091, +0.07080]$. The identical-control scores differed at 2 of 220 pairs. Treatment scores agreed with the next control or terminal walk at 199 of 207 adjacent comparisons. These diagnostics do not change the primary result. The experiment establishes neither agreement within nor disagreement beyond the specified tolerance on airline. The separate first-point contrasts on airline are unchanged.

Separate full technical replication. After the recovered attempt and its result had been inspected, we ran one fresh serial campaign over the same 50 airline tasks and three outer seeds. We retained the same actor, customer, policy, scorer, E7 comparator, estimator, bootstrap seed and ± 0.10 tolerance. All 150 task-seed units completed without retry. All 819 clones completed and 102 measured points were informative. This repetition is a technical check on the original task pool, not a new sample of independent tasks. It is reported separately from the recovered attempt.

The primary residual is $+0.00667\,[-0.06000, +0.08000]$ over 150 units from 50 tasks. Its task-clustered 95% interval is wholly within the fixed tolerance band. Under the registered rule, R_{occ} is within tolerance and P is unresolved. The count-scaled error is $-0.02667\,[-0.15333, +0.11333]$ on the same 150 units. The horizon contrast remains too wide at $+0.00000\,[-0.13580, +0.13750]$ over 85 units from 35 tasks. These results do not resolve first-point substitution or the horizon mechanism.

The secondary within-campaign decomposition qualifies the primary result. On the same 150 units, the scheduled residual is $-0.02667\,[-0.05333, -0.00667]$ and the between-campaign component is $+0.03333\,[-0.03333, +0.10667]$. Identical execute clones agree at all 273 pairs, while treatment scores disagree with the next execute clone or terminal walk at four of 273 adjacent comparisons. The former checks local score stability. Neither check proves that every part of a cloned state and its future trajectory is identical. The 50 tasks and the E7 policy effects are shared across both E10 attempts, so their intervals are not independent confirmations. The original count-only deviation remains attached to the recovered attempt.

Exploratory path follow-up. After inspecting the four adjacent disagreements in the technical repeat, we selected those four task-seed units for three additional technical runs each. The 12 runs visited 17 mutating points and completed 51 clones without a capture failure. Three of 17 adjacent reward comparisons disagreed, while no execute/execute reward pair did. In each disagreement, the recorded starting messages, pending proposal, budget and horizon matched. The first observed path divergence was the actor returning different text for identical recorded request JSON at seed zero and temperature zero. All 51 clones matched their parents on 12 captured state components and the availability map. Two execute/execute pairs also differed in actor text despite equal rewards. This follow-up was selected after the repeat's results were known and uses the same tasks. It does not identify a backend cause or explain the original four disagreements.

X THE MEASUREMENT FLOOR

At 81.1% of E6's 270 branch points no tested arm scores above zero, so the tested arms cannot differ there. On the first instrument the figure was 87.8% of E2's 434 points, and on airline it is 53.0% of E7's 115, with informative points on 20 tasks. The floor is concentrated rather than diffuse.

Adding outer seeds under greedy decoding does not add independent tasks. In the fixed-budget 32B probe, the number of capable tasks stays at 21, but only 11 tasks overlap the 14B capable set (Appendix X.1). The 32B actor uses more calls and more often exhausts its budget before a mutating

Table X.2: The 32B probe against the 14B baseline, 342 episodes each, none failed. The first row determines the pre-registered outcome, the pool label. The other rows are descriptive, and the budget rows were computed after the registered result was known.

	14B	32B
capable tasks, of 114	21	21
of which capable at the other size	11	11
capable in every seed	14	12
mean success	0.152	0.149
out of budget, of 342 episodes	23	173
before any mutating proposal	5	148
mean tool calls per episode	6.16	8.49
mean mutating opportunities per episode	1.77	1.24

proposal. These observations do not isolate model scale from the task pool or horizon. A wider pool or a second domain can test whether the floor persists.

X.1 Experiment 5: The 32B Actor Leaves the Capable-Task Count Unchanged

E5 asks whether the floor in Appendix X reflects the 14B actor or the task pool at this budget. It runs the unguarded policy with the E4 runner on the same 114 retail tasks, seeds and budget, with Qwen2.5-32B-Instruct-AWQ as the actor. The simulated customer remains on the 14B model. An amendment made before any 32B episode pinned the customer model. The runner had used the actor's model for the customer, so a larger actor would otherwise also have been a more capable customer.

A task is capable when the unguarded policy succeeds under at least one seed, and the 14B baseline of 21 capable tasks was fixed from E4's unguarded arm before the probe ran. The registered rule assigns the actor label if the 32B count is at least 42 and the pool label if it is at most 31. These are decision labels rather than causal diagnoses.

The 32B actor is capable on the same number of tasks, exactly 21, but only 11 overlap. The capable count therefore stays fixed in this probe while membership changes. The larger actor exhausts its ten calls more often, including 148 episodes before a mutating proposal. It also makes more tool calls on average. These observations are consistent with budget pressure but do not isolate it as the cause of the unchanged capable count. The probe measures unguarded success, not guard value, and the branch experiments were not run at 32B.

Y Deviations and Corrections

This section lists the corrections and amendments that bear on the reported results. Amendments specific to E8 are listed in Appendix W.

Correction 1: the branches stopped at the agent's first message to the customer. The E1 to E3 continuation loop ended a branch when the actor replied in text. In this benchmark, that reply is addressed to the simulated customer. The loop that runs whole episodes hands that reply to the customer and carries on. Those branches were therefore scored at the agent's next message, not at the episode end specified by Equation (1). In E2's records 91.9% of allow continuations end that way, a mean of 1.44 steps after the branch point, while 93.3% of E4's whole unguarded episodes end when the conversation does. E1 to E3 are excluded (Appendix S). E6 replaces E2 and was registered after the defect was found. Whole episodes and branches now use the same loop. The same records showed that a refusal the agent is not told about appends nothing to the conversation. In E2, 513 of 868 paired block continuations took exactly one more step than their allow twin, consistent with a one-step delay, although these records do not identify the next proposed action.

Correction 2: a pooled test that averaged gaps of opposite sign. E7's rule for the noticed refusal pooled its signed first-point gap over retail and airline. The pre-registration did not anticipate that the

policy's effect would change sign between domains, although the audit had already shown it does. The opposite-signed gaps canceled, so the pooled rule reported agreement. The same pooling gives silent refusal an interval containing zero although its gap excludes zero in each domain separately (Appendix V). The pooled test's output is reported as computed. The per-domain gaps, listed among the pre-registered reported quantities, provide the result. Retail's noticed-refusal gap meets the registered width rule, which is not an equivalence test, while airline's does not.

Amendment: E7's replication rule made two-sided. E7's replication rule was first written to count only a gap below zero, the side retail's gap fell on. E6's own rule was two-sided, and airline is where the audit found blocking helps. The rule was made two-sided before the full run, when one excluded test unit had shown a positive gap.

Instrument defects in record keeping. An exception handler recorded episodes that never ran as having zero opportunities and zero elapsed time, and a late exception in an episode discarded the decision points it had already measured. The affected records are excluded. Both defects were fixed and are covered by regression tests.

Z Clone Conformance and Reproducibility of the Branch Experiments

The branch experiments run on the retail and airline pools of the same benchmark as the audit, 114 and 50 tasks, with the same 14B open-weight actor served locally and the same interaction budget of ten. All reported runs use the same fixed engine configuration. The offline package omits private run-control metadata and retains the inputs needed by its documented analysis commands.

The audit and branch experiments ran on different hosts and serving stacks (Appendix O). The audit half was served by vLLM 0.24.0 on 46 GB GPUs. The branch half was served by vLLM 0.8.5.post1 on 80 GB GPUs, with the PyTorch, transformers and CUDA versions shown in Table O.1. Both halves use the same fixed checkpoint and benchmark revision. Every comparison this paper makes is within one half, so no estimate is differenced across the two stacks. E6 ran on two identically configured 14B servers on one host, with each worker's actor and customer on the same server as in E4. E5 served the 32B actor on one GPU and kept the customer on the 14B server E4 used, and its report records the model of its actor and of its customer separately.

Each experiment writes one record per sampled decision point, or per unit containing all arms for E6 and E7. Records are flushed after every episode, preserving everything already measured if a run stops prematurely. Each record carries the proposed action, the engine's decision, whether the decision changed the action, every continuation's reward and stop reason for every arm, and whether the point was informative. The offline analysis scripts consume the released records and reference reports. Registered analyses and later exploratory analyses are distinguished in Appendix R.

Failures are recorded as failures. A field that was not measured is null rather than zero, and aggregates exclude failed episodes rather than averaging a constant into them. Table Z.1 lists every check on the clone path, and the subsections below describe them.

Scheduling, repetition and the cost of a check. The audit interleaves its arms by task to spread any drift in the served model across arms, although drift was not independently ruled out. A branch experiment runs one unit's arms consecutively from the same prefix, which is what makes them paired. E4 through E8 each used one campaign, with re-runs only for failed units. E10 also has the separately reported technical repeat. Variation across seeds is within tasks, not across independent campaigns. The audit and branch experiments charge interventions differently, which matters when comparing their results. In the audit a guard check costs one interaction slot whether or not it changes the action, because that is the resource the arms are matched on. In the branch experiments, a refusal executes nothing and charges no slot but consumes a step. The reviewed engine is the exception, since its review is a model call and is charged.

Table Z.1: Clone conformance checks. The mock and retail fixtures replace model text with fixed replies and compare a cloned continuation with an uninterrupted run, while the telecom suite tests state copy, branch independence, message replay and budget forks separately. Live checks use the served 14B actor and customer on airline. An adjacent comparison sets a treatment clone against the guarded walk's next segment followed by the next execute clone, which must agree if cloning and serving are exact. The deterministic server disables prefix caching and serves one sequence at a time. Path agreement compares every request and response after renaming tool-call identifiers. Every deterministic clone comparison reached the same reward.

Check	Compared	Result
Mock loop, four guard policies	actions, budget, reward	equal
τ^2 telecom, solo environment	database, messages, budget	23/23 tests
τ^2 retail, fixed text	customer turn, pending write	2/2 cases
Live state copies, follow-up	12 state components	51/51
Live execute pairs, technical repeat	terminal reward	273/273
Live adjacent pairs, technical repeat	terminal reward	269/273
Request replay, production server	re-send against in-run recording	14/444 differ
Request replay, both servers	two re-sends against each other	0/444 differ
Deterministic re-run, 4 keys $\times$ 3	execute, adjacent paths	33/33, 33/33
Deterministic run, 50 airline tasks	execute, adjacent paths	91/91, 91/91
Deterministic run, second campaign	execute, adjacent paths	90/91, 90/91
Same regime, sum and whole episodes	tasks with equal effect	47/50
Same regime, engine, synchronized opening	tasks with equal effect	50/50
Deterministic runs, later studies	execute-pair reward	940/940

Z.1 Deterministic Conformance Fixtures

Since E6, whole episodes and branch continuations run one loop. Deterministic mock tests compare prefix plus continuation against uninterrupted execution for four policies, including actions, budget use and reward. A separate offline suite passes 23 tests in the pinned τ^2 telecom solo environment, checking database copy, branch independence, message replay, fixed-action continuation, budget forks and database-state equality. One deterministic retail case also matches an uninterrupted customer continuation after a prefix clone, with generated customer text fixed. A second deterministic retail case matches a state-writing pending proposal, budget ledger, customer continuation and nonzero terminal score across an unbranched run and a cloned continuation. These checks do not establish full conformance under live actor and customer generation.

Z.2 Serving Determinism and Deterministic Replay

The airline technical repeat left four adjacent reward disagreements and a within-campaign residual of $-0.02667\,[-0.05333, -0.00667]$ (Appendix W.1), and the selected follow-up found different actor text for identical recorded requests. A pre-registered diagnostic separates serving variation from copying error. It computes no effect estimate and is not merged with any experiment. Two servers host the same 14B model on one GPU. The production server keeps the configuration of every branch experiment, with automatic prefix caching and continuous batching. The deterministic server disables prefix caching and serves one sequence at a time. The memory setting was amended before any output because the serving library counts other GPU processes against the requested memory fraction. Responses are compared by message text, tool-call names and arguments, and finish reason. Paths are compared request by request and response by response after tool-call identifiers are renamed consistently, and this comparison detects a deliberately altered real request.

Request replay. The 444 actor and customer requests recorded in the 12 follow-up runs were each sent twice, one at a time, to each server. On both servers the two re-sends agreed for every request, which meets the registered criterion for the deterministic server. On the production server, 14 of the 444 re-sent requests, 9 from the actor and 5 from the customer, returned a response different from the one recorded during the run, and 3 of these changed a tool call. The production server's output for a request is therefore not a function of the request alone. The deterministic server's responses

Table Z.2: Clone comparisons on the production and deterministic servers. The first two columns summarize three runs each of the same four airline task-seed units. The third runs every airline task once. Visited points differ between servers because their responses differ. A repeat walk compares a unit's later run with its first run. An empty opening means the actor's first request carried an empty user message.

	production, 4 keys	deterministic, 4 keys	deterministic, 50 tasks
runs completed	12/12	12/12	50/50
visited points	17	33	91
clones completed	51/51	99/99	273/273
execute pairs, same path	15/17	33/33	91/91
execute pairs, same reward	17/17	33/33	91/91
adjacent pairs, same path	12/17	33/33	91/91
adjacent pairs, same reward	14/17	33/33	91/91
repeat walks, same path	4/8	6/8	–
runs with an empty opening	1/12	2/12	0/50

differ from the production recordings for 52 requests, as expected from its different computation path.

Deterministic re-runs. Table Z.2 compares the production follow-up with the same keys on the deterministic server. On the production server, 5 of 17 adjacent comparisons and 2 of 17 execute pairs followed different paths, and in each case the first difference was a response to an identical request. On the deterministic server, all execute pairs and adjacent comparisons followed identical paths and reached the same reward, 33 of each. On the deterministic server, 37 of the 50 airline tasks reached a mutating proposal. Over 273 completed clones at their visited points, all 91 execute pairs and 91 adjacent comparisons followed identical paths and reached the same reward.

The two repeat-walk differences on the deterministic server, like two of the four on the production server, begin at the actor's first request, whose opening user message was empty. The benchmark's reset had returned before the simulated customer's opening message was available. This occurred only at the start of an episode, never inside a clone, which resumes from a recorded decision state.

Outcome. The diagnostic met the registered criterion, and no tested clone comparison differed. Under deterministic serving, the copy therefore reproduced the recorded continuation at every tested point. This is consistent with serving variation as the source of the technical repeat's adjacent disagreements. It does not establish the cause of the original four, which cannot be replayed on the same server state, and it does not establish equality of latent state beyond the recorded requests, responses, databases and scores. The repeat-walk differences begin in a request after matching responses. The registered rule counts this as state disagreement where agreement should hold, but the state in question is the episode start, not the copy. Comparisons between separately started episodes, including the policy-effect campaigns against which the occupancy sums are paired, therefore carry this variation as well as serving variation, and no general reliability claim is made for them.

Z.3 Same-Regime Occupancy Check

A second registered study asks whether, within one serving regime and on the complete airline pool, the occupancy sum on guarded walks matches the policy effect measured by separately started whole episodes. The deterministic server is restarted with the same command. Three new sets cover all 50 airline tasks at outer seed 0, with every model call recorded. They comprise guarded walks with every visit cloned, unguarded whole episodes and noticed-refusal whole episodes. The adapter's reset race is deliberately left in place so that the result applies to the harness that produced the other experiments. For task i, S_i is the walk's occupancy sum and D_i the guarded episode's terminal reward minus the unguarded one's. The primary quantity is $R_{\mathrm{occ}} = \mathrm{mean}_i(S_i - D_i)$ with the registered task bootstrap and the E10 tolerance of ± 0.10.

Table Z.3 gives the result. The residual is $-0.02000\,[-0.10000, +0.04000]$, within the registered band with its lower bound at the band's edge. The occupancy sum equals the separately measured effect in 47 of the 50 tasks. In each of the other three, the unguarded episode began with an empty

Table Z.3: The same-regime check on all 50 airline tasks, outer seed 0, deterministic server. S_i is the walk's occupancy sum, D_i the effect from separately started episodes and W_i the walk's own effect, its reward minus the first-visit execute clone's. Paths are compared after renaming tool-call identifiers. Interval is a task bootstrap 95% interval.

runs completed (walks, unguarded, guarded)	50, 50, 50
tasks with a visited point, visited points, clones	37, 91, 273/273
R_{occ}	$-0.02000\,[-0.10000, +0.04000]$
$K\Delta_1 - D$	$+0.02000\,[-0.14000, +0.18000]$
tasks with $S_i = D_i$	47/50
tasks with $S_i = W_i$	50/50
execute pairs, same path and reward	90/91, 91/91
adjacent pairs, same path and reward	90/91, 91/91
guarded episode against walk, same path	49/50
unguarded episode against first-visit execute clone, same path	47/50
new walks against the earlier deterministic walks, same path	50/50
runs with an empty opening (walks, unguarded, guarded)	0, 3, 1

Table Z.4: The reviewed engine on all 50 airline tasks, outer seed 0, deterministic server with synchronized openings. S_i, D_i and W_i are as in Table Z.3. A budget-stopped review consumed the last interaction slot, so the proposal did not run. The rows for D, Δ_1, execute pairs and review outcomes are descriptive and were added after the result was known. Intervals are task bootstrap 95% intervals.

runs completed (walks, unguarded, engine)	50, 50, 50
tasks with a visited point, visited points, clones	37, 90, 270/270
R_{occ}	$0.00000\,[0.00000, 0.00000]$
$K\Delta_1 - D$	$-0.04000\,[-0.10000, 0.00000]$
D	$+0.06000\,[0.00000, +0.14000]$
Δ_1, all units	$+0.02000\,[0.00000, +0.06000]$
tasks with $S_i = D_i$, with $S_i = W_i$	50/50, 50/50
execute pairs with the same reward	90/90
visits by review (allow, block, replace, stopped by budget)	51, 19, 13, 7
visits with nonzero value, same order	0, 0, 2, 1
resets repeated for an empty opening	1 of 150

opening user message, so it started from a different state than the walk. Within every walk the sum equals the walk's own effect, and every new walk repeats the earlier deterministic walk of the same task exactly.

Two clone comparisons in this campaign, one execute pair and one adjacent pair, followed different paths without changing the reward. In both, starting messages, proposal and budget matched. The first difference was an observation returned by the benchmark's threaded episode loop. One execute clone received a tool result and the other a customer message. The adjacent pair received different customer messages. Under the registered rule, within one deterministic serving regime the occupancy sum matches separately measured policy effects except where the episode start differs. This is consistent with the earlier airline residuals reflecting differences between campaigns and serving regimes rather than the instrument. The count-scaled first value remains unresolved.

Z.4 Reviewed Engine Under Deterministic Serving

A third registered study, written before any of its runs, applies the same check to the reviewed engine, a selective guard that charges one interaction slot for every review. On the deterministic server it runs engine walks with every visited point cloned, and unguarded and engine whole episodes, on all 50 airline tasks at outer seed 0. Unlike the same-regime check, a reset whose opening user message is empty is closed and repeated, at most five times, with every attempt logged. An allow is measured like any other visit, because the review that produced it has already charged a slot.

Table Z.4 gives the result. The residual is zero, and the occupancy sum equals the separately measured effect on every task, including the 13 without a visited point. The point estimates of the first

Table Z.5: Silent refusal with every visit measured, on the 25 informative retail tasks, outer seed 0, deterministic server with synchronized openings. The schedule rows re-weight the same walks by the registered four-point schedule. Rows below the schedule rows are descriptive and were added after the result was known. Intervals are task bootstrap 95% intervals.

runs completed (walks, unguarded, silent)	25, 25, 25
tasks with a visited point, visits, clones	24, 549, 1,647/1,647
R_{occ}, every visit	$0.000\,[0.000, 0.000]$
tasks with $S_i = D_i$, with $S_i = W_i$	25/25, 25/25
schedule sum minus every-visit sum	$+0.08\,[0.00, +0.20]$
R_{occ}, fixed schedule	$+0.08\,[0.00, +0.20]$
tasks where the two sums differ	2 of 25
D	$-0.24\,[-0.48, 0.00]$
Δ_1, all units	$0.00\,[0.00, 0.00]$
$K\Delta_1 - D$	$+0.24\,[0.00, +0.48]$
tasks with a nonzero visit, at the last visit	10, 8
execute pairs with the same reward	549/549
clones per task, every visit and schedule	65.9, 17.3
resets repeated for an empty opening	0 of 75

value and of the first value scaled by the visit count both fall below the effect of $+0.06$. Where the review allowed the proposal, at 51 visits, the one-step value was zero. At seven visits, review consumed the last interaction slot and prevented the proposal from running. At one visit, this changed the outcome, giving a charged review nonzero value without an override. The two other nonzero values come from replacements. Under the registered rule, the occupancy sum matches the separately measured effect of this budget-charged selective guard within tolerance, and here exactly.

Z.5 Silent Refusal With Every Visit Measured

A separate registered check measures silent refusal at every visit, the regime E8 approximated with a fixed schedule. It uses the 25 retail tasks that carried every informative point of E6, a list fixed before E8, at outer seed 0. It runs silent-refusal walks with every visited point cloned, and unguarded and silent whole episodes. For throughput, the study used a third serving stack, three identical deterministic servers of vLLM 0.11.0 on one GPU. Model weights were identical file by file to the other runs' copy, and openings were synchronized. Each task's walk and whole episodes ran on one server. The primary quantity is $R_{\text{occ}} = \text{mean}_i(S_i - D_i)$ under the ± 0.10 band. The secondary analysis applies the registered schedule to the same walks. It takes the first and last visits, the two specified interior positions for the walk's own K, and the margin from $K - 2$, weighted as in E8. Three walks completed on the first stack before the move are reported only as a cross-stack comparison.

Table Z.5 gives the result. With every visit measured, the occupancy sum equals the separately measured effect on every task, and the residual is zero. Ten tasks have a nonzero one-step value at a single visit each. Eight occur at the last visit and two at interior visits omitted by the schedule. The schedule applied to the same walks misses exactly those two, so its sum differs from the every-visit sum on two tasks and its residual crosses the band. Under the registered rule, the unresolved E8 silent result reflects the fixed schedule rather than the instrument, on this informative subset. One silent refusal has no value on any task, and the count-scaled first value misses the whole effect. The three walks run on the first stack give the same occupancy sums and walk rewards as their counterparts here, with slightly different visit counts.

Z.6 A Second Actor

A further registered check measures a second actor. Qwen2.5-7B-Instruct serves as both actor and simulated customer, the convention of the branch experiments, on all 50 airline tasks at outer seed 0, with openings synchronized. Four identical deterministic servers of vLLM 0.24.0, the audit's release, carry three sets, namely branch arms at the first mutating proposal as in E7, walks with every visited point cloned under noticed refusal and under the engine, and whole episodes under no guard, silent refusal, noticed refusal and the engine. The primary quantity is the silent first-point gap

Table Z.6: The second actor, Qwen2.5-7B, on all 50 airline tasks, outer seed 0, deterministic serving. Per-point quantities use the 44 branch points. First-point and count-scaled values come from the branch records and average over all 50 units with zero where no mutating proposal occurs. D is the whole-episode effect against no guard. Intervals are task bootstrap 95% intervals.

	silent	noticed	engine
$D - \Delta_1$	$+0.227\,[+0.114, +0.364]$	$+0.182\,[+0.068, +0.295]$	$0.000\,[0.000, 0.000]$
interventions, always arm	28.77	4.43	0.95
D	$+0.20\,[+0.10, +0.32]$	$+0.20\,[+0.10, +0.32]$	$+0.02\,[0.00, +0.06]$
Δ_1	$0.00\,[0.00, 0.00]$	$+0.04\,[-0.04, +0.12]$	$+0.02\,[0.00, +0.06]$
$k\Delta_1$	$0.00\,[0.00, 0.00]$	$+0.26\,[-0.12, +0.74]$	$+0.06\,[0.00, +0.18]$
occupancy sum, mean	not walked	$+0.20$	$+0.02$
R_{occ}	not walked	$0.000\,[0.000, 0.000]$	$0.000\,[0.000, 0.000]$
tasks with $S_i = D_i$	not walked	50/50	50/50
visits, clones	not walked	165, 495/495	136, 408/408
execute pairs, same reward	not walked	165/165	136/136

Table Z.7: Post hoc cross-fitted guard selection on the second actor's 50 airline tasks, 1,000 random half splits used in both directions. Regret is in task success on the scoring half. Choice shares are over the 2,000 choices. Tokens are actor tokens per choosing task, in thousands.

Selector	Mean regret	Share with regret	No guard	Engine	Noticed	Tokens (k)
first-point value	0.1070	0.5000	0.1560	0.3440	0.5000	199.6
count-scaled value	0.0486	0.2060	0.0845	0.1215	0.7940	384.5
occupancy sum	0.0000	0.0000	0.0000	0.0000	1.0000	899.2
whole episodes	0.0000	0.0000	0.0000	0.0000	1.0000	288.2

per branch point under the registered E6 rule. The occupancy residuals and the ranking of the four policies by each substitute, against whole episodes, are secondary. The per-unit count-scaled error in Table R.1 and the execute-pair counts were added afterward.

Table Z.6 gives the result. All 50 branch units completed, 44 reaching a mutating proposal, and one reset in 350 needed a repeat. The registered silent test replicates for the second actor. One silent refusal has value zero at every branch point while refusing throughout gains $+0.227$, so the gap excludes zero, in the direction of the 14B airline result. The noticed gap also excludes zero here. For both walked policies the occupancy sum equals the separately measured effect on every task. By whole-episode point estimates, silent and noticed refusal tie at $+0.20$, followed by the engine at $+0.02$ and no guard at zero. The first-point and count-scaled substitutes rank silent refusal level with no guard and below the engine. The engine's effect is a tenth of silent refusal's, with non-overlapping intervals, so the ranking error does not depend on the tie between refusals. The occupancy sum, measured for noticed refusal and the engine, keeps their magnitude and order.

Choosing a guard on held-out tasks. A post hoc analysis specified after these records were examined measures the cost of selecting a guard on some tasks and using it on others. Each of 1,000 random splits divides the 50 tasks into halves of 25, each used once for choosing and once for scoring. A selector scores each candidate by its mean over the choosing half, with no guard at zero and ties favoring the less intrusive candidate. Regret is the best candidate's mean whole-episode effect on the scoring half minus the selected candidate's. Silent refusal was not walked, so the occupancy sum chooses among the other three. Cost is the actor tokens that each selector's inputs consumed per choosing task, counting a branch prefix once and each walk as its guarded episode plus its clones. Table Z.7 gives the result. The first-point value chooses the engine or no guard in half of the 2,000 choices, and the count-scaled value in a fifth, while occupancy sums and whole episodes always choose noticed refusal, whose pooled whole-episode effect equals silent refusal's. Whole episodes use about a third of the walks' tokens.