CORRECTION HAS A WINDOW: HOW LONG A FALSE PREMISE STAYS FIXABLE IN TOOL-USING AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

A tool-using agent reads a record, the record is wrong, and later the truth arrives. We ask how much the timing of that correction matters. We overwrite one field in a tool result that a τ-bench retail agent has already read, let the agent act on it for j live turns, and then delete the false record and append the true one. Across ten policy families from five vendors (30,942 episodes with a simulated customer), when the false value licenses a refusal, a correction one turn in buys back 0.678 $[0.602, 0.754]$ of the task success the false premise cost and four turns in only 0.254 $[0.132, 0.376]$ (pooled over the nine families the fault damages, on the same units). This decline, the *correction window* (success corrected at $j = 1$ minus $j = 4$), is positive in all six API-served families, where it pools to $+0.287$ $[+0.217, +0.358]$ and to $+0.294$ $[+0.210, +0.377]$ net of the same edit applied to a context with nothing false in it; net of that edit, none is detected in the four open-weight families we served locally. Withholding the same record until the same turn shows that lateness is part of the window but not all of it. Among episodes that reach the late correction the window is smaller under a second simulated customer, though not when every scheduled episode is counted. Late resets of the agent's side do not reliably restore success, and an explicit retraction adds no reliable benefit over silent correction. All code, data and episode traces are in the supplementary material and will be released at `https://github.com/xxx/xxx` upon acceptance.

1 INTRODUCTION

Agents that call tools spend most of their turns reading. A customer-service agent on τ-bench (Yao et al., 2025) looks up a user, an order and a product before it acts, and every one of those observations is trusted on arrival. Some are wrong: a cache serves a stale row, an upstream service returns a plausible default, a record is edited between reads. Tool-using agents fail silently (Advani, 2026; Panthi & Abdelfattah, 2026) and adopt corrupted tool output rather than question it (Yang et al., 2026; Xu et al., 2026). What has not been measured, for a false rather than a missing record, is the quantity a system designer needs: once an agent has read something false, *for how long does supplying the truth still work?*

We build an instrument on τ-bench retail to measure it, in which one field of one observation inside a nine-step read-only investigation prefix is rewritten; after the agent has acted on the false premise for $j \in \{1, 2, 4\}$ live turns (a live turn is one agent action: a tool call or a message to the customer) we *withdraw* the observation from the agent's context, *correct* it (deleting the false record and appending the true one as a fresh tool return), or *retract* it (correcting and telling the agent an earlier observation was wrong). Further arms ask what exactly is lost: one withholds the true record until turn j instead of falsifying it, one swaps the simulated customer, and others rewind the agent's transcript or reset the database at the moment of correction.

Of ten policy families from five vendors (six served by their providers' APIs, four open-weight models served locally; Table 1), in the nine the fault damages the same correction buys back 0.678 $[0.602, 0.754]$ of the damage one turn in and 0.254 $[0.132, 0.376]$ four turns in, on the same units. In success points the window ($j = 1$ minus $j = 4$) is positive and signif-

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.

icant in every API-served family (four of the six after a Holm correction for ten) and pools there to $+0.287$ $[+0.217, +0.358]$, or $+0.294$ $[+0.210, +0.377]$ net of the identical edit on a clean prefix. Pooled over the four open-weight families served locally it is not detected ($+0.007$ $[-0.106, +0.120]$): the fault barely damages `Granite-3.3-8B`, on the two 14B `Qwen` models a correction recovers as much late as early (`Qwen3-14B` at 4 or 8 bits), and `Qwen3-30B-A3B`'s raw window ($+\mathbf{0.126}$ $[+0.015, +0.237]^*$) is matched by the same edit on a clean prefix.

Lateness accounts for part of the window. Gulati et al. (2026) showed that ground-truth *clarifications* injected later into a long-horizon trajectory help less; nothing false is believed there. Withholding the true record until turn j instead of falsifying it (five API-served families), the window is $+0.129$ $[+0.071, +0.187]$ against $+0.309$ $[+0.220, +0.398]$ for the false premise on the same units; the excess, $+0.180$ $[+0.082, +0.279]$, bounds from below what believing a false version costs beyond lateness (§3.2).

The width also depends on what the false value licenses. On a ladder of *inert*, *mild* and *blocking* values for the same record, the same unit's window is $+0.206$ $[+0.097, +0.314]$ wider when the false value licenses a refusal no tool call contradicts than when it leaves the agent attempting operations the environment rejects, though the refusal also does more damage. A second simulated customer shrinks the window among episodes that reach the late correction but not when every scheduled episode is counted, and late resets of the agent's side do not reliably restore success (§3.6). At $j = 4$ a notice telling the agent to distrust its own earlier statements adds $+0.036$ $[-0.064, +0.136]$ to silent correction, no more than a message restating the rule to confirm before acting.

Contributions. (i) An interventional instrument for how long a false premise stays correctable, with controls and tests, three of which (a placebo, an equivalence test and a floor check) overturned conclusions we would otherwise have drawn (Appendix B). (ii) The correction window in ten families: significant raw in all six API-served ones (four net of the edit's cost), and net of it detected in none of the four served locally (§3.1). (iii) A lower bound separating lateness from falsity, using an absent record withheld until the same turn (§3.2). (iv) Evidence that the window is wider on the rung whose false value licenses a refusal, also the more damaging one (§3.5), and tests of where the loss lives with three customers and five late edits (§3.6). (v) A placebo-controlled test of explicit retraction, which shows no reliable benefit (§3.4).

2 THE INSTRUMENT

Environment and policies. We use τ-bench retail (Yao et al., 2025): an agent with database tools serves a simulated customer, and an episode succeeds only if the final database state and required outputs match the ground truth. The customer is `gpt-4o-mini` at temperature 0 in every API-served arm, the `gpt-4o-mini` policy's included (§3.6 swaps it). Ten policy families run in live rollouts (Table 1): six served by their providers' APIs and four open-weight models served locally with vLLM on one GPU (30% of episodes). `Qwen2.5-14B-Instruct-AWQ`'s customer is by default the same local model (§3.8 swaps it); `Qwen3-14B-AWQ`, `Qwen3-30B-A3B-GPTQ-Int4` (a mixture of experts, 3B active) and `Granite-3.3-8B-Instruct` have the API-served one. Four further models were dropped at or before the screen (Appendix B). Tasks are screened per policy for clean solvability at a seed no analysis reads and re-run independently, so the no-fault baseline is 0.54–0.85, not the 1.0 of the screen.

The false premise. Every episode opens with a nine-step *read-only investigation prefix*: the user, order and product lookups the task requires, executed against the real environment and replayed as the agent's own past tool calls after the customer's opening message. Because read-only calls leave the database unchanged, the prefix can be permuted to place any observation k reads from its end with content and length unchanged, and a no-fault control exists at every arrangement used, since permutation alone moves accuracy by up to 0.29. The fault rewrites *one field of one observation*: the `status` of the order the task turns on. Faults are written into the pre-built prefix, never through a wrapper on `env.step` (Appendix A.1). A second fault family (§3.7) rewrites a product-record field read by a different tool.

Table 1: Policy arms. Clean accuracy is the independent re-run on each arm's own solvable task set; cost is API spend on the policy and the customer (for the local arms, the API-served customer alone). Episodes include the seed-0 screens, the product family, the airline port and the customer swaps. The `Qwen3` models run with thinking off; the last row is the 8-bit re-serving of §3.1, on `Qwen3-14B-AWQ`'s units.

policy	serving	vendor	clean acc.	episodes	cost
`gpt-4o-mini`	API	OpenAI	0.744	9,778	$141.23
`deepseek-v4-flash`	API	DeepSeek	0.848	4,524	$68.43
`gpt-5-mini` (reasoning on)	API	OpenAI	0.810	3,428	$99.28
`qwen-plus`	API	Alibaba	0.690	2,263	$100.36
`gpt-4.1`	API	OpenAI	0.810	927	$140.99
`doubao-seed-2.0-lite`	API	ByteDance	0.696	627	$8.68
`Qwen2.5-14B-Instruct-AWQ`	local vLLM	Alibaba (open)	0.538	5,320	$3.35
`Qwen3-14B-AWQ`	local vLLM	Alibaba (open)	0.577	1,443	$1.33
`Qwen3-30B-A3B-GPTQ`	local vLLM	Alibaba (open)	0.701	1,431	$1.34
`Granite-3.3-8B-Instruct`	local vLLM	IBM (open)	0.608	535	$0.49
`Qwen3-14B-FP8` (re-serving)	local vLLM	Alibaba (open)	—	666	$0.60

The ladder. The same record on the same units is rewritten three ways. *Inert*: an item detail the task never touches. *Mild*: `status delivered`→`pending`, which leaves the order modifiable in principle but makes the agent's attempts fail. *Blocking*: `delivered`→`cancelled`, which licenses the agent to decline, and no tool call contradicts a refusal.

The interventions. The corrupted observation sits at $k = 0$, the last read before the first live turn, in every window arm. After $j \in \{1, 2, 4\}$ live turns ($j = 8$ is exploratory), the harness performs one of: **withdrawal** (T5), deleting the observation; **correction** (T6), deleting it and appending the true record as if just re-fetched (an append-only variant keeps the false record); **retraction** (T7), correction plus a harness notice that names the record, says an earlier observation was incorrect, and tells the agent to distrust its own earlier statements about it (Appendix A.3); and **absence** (T10), in which the target read is replaced by a read of a related entity and the true record arrives at turn j, either free to be fetched by the agent in the meantime or **withheld** (T11: the harness answers every earlier read of it with "the order service is temporarily unavailable"). Five further edits accompany the correction at $j = 4$: a **rewind** of the agent's live transcript to the prefix (T12), a **scrub** of the agent's replies to the customer (T12), and a **reset** of the database to its state at the end of the prefix (T13), alone, with the rewind, or with the rewind and the customer simulator returned to its opening message (a *full reset*); otherwise the customer keeps its history, and each edit is also applied to a clean prefix. Two controls accompany them: the **surgery control** (T8) performs the identical deletion and insertion at the identical turn on a prefix with nothing false in it, and **placebo notices** are same-channel messages at the same turn with no retraction content. The correction arm exists because conditioning on whether the agent *spontaneously* re-read the record is post-treatment (the fault and the lag cause both the re-read and the outcome), so the effect of supplying the true record is identified by supplying it *for* the agent.

Statistics. The unit is a task; every per-family estimate is a paired bootstrap over units (Miller, 2024) with 95% percentile intervals from $8{,}000$ resamples. A unit's value in a cell is its mean over the seeds shared by every cell of the comparison, and a lagged cell keeps only episodes that reached the intervention (Appendix A.6 gives retention and an intention-to-treat check). Trends in j use a one-sided within-unit permutation test on the lag labels ($5{,}000$ permutations). Equivalence claims use two one-sided tests at $\delta = 0.20$, fixed before the tests were run; a 90% interval said to lie inside ± 0.10 describes its width, not a second margin, and other nulls are reported with that interval, not as equivalence. * marks an interval excluding zero (on a trend, $p < 0.05$); per-family stars are uncorrected, and §3.1 also counts windows and trends after a Holm correction. Pools are DerSimonian–Laird random effects (DerSimonian & Laird, 1986) on the bootstrap standard errors, with Cochran's Q and I^2; with two to ten families these under-cover, so we also give a modified Hartung–Knapp interval (Hartung & Knapp, 2001; Röver et al., 2015) (Appendix A.6). The share of the damage a correction recovers is $(c_j - f)/(b - f)$, with c_j the success corrected at lag j, f under the never-corrected fault and b under the no-fault control, on the units all four cells share.

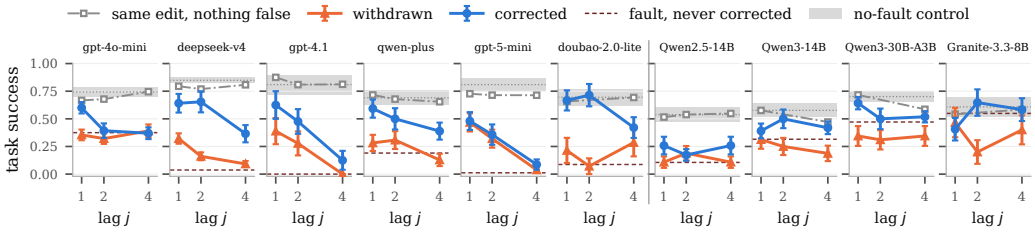


Figure 1: Task success after a correction (●) or a withdrawal (▲) applied j live turns after the agent started acting on a blocking false premise. Grey band: no-fault control ± s.e., dotted at its mean; grey dash-dot (□): the correction's surgery control, the identical deletion and insertion on a clean prefix. Left of the rule, the six API-served families (`deepseek-v4-flash` and `doubao-seed-2.0-lite` abbreviated); right, the four served locally. At $j = 1$ and $j = 4$ the corrected and grey points are on the units of the surgery-adjusted contrast (Table 2); each curve's other points are on the units its own cells share. Error bars: s.e. over units.

Table 2: The correction window on the blocking rung: success corrected at $j = 1$ minus at $j = 4$, paired on units; the within-unit slope per turn over $j \in \{1, 2, 4\}$, on the units with all three lags, with its one-sided permutation p (0.0002 is the smallest attainable); and the window minus the same contrast on the surgery control, on the units the four cells share. *: interval excludes zero.

policy	window (j=1 − j=4)	trend (p)	surgery-adjusted
`gpt-4o-mini`	**+0.230** [+0.133, +0.339]*	−0.071* (0.0002)	**+0.309** [+0.173, +0.452]*
`deepseek-v4-flash`	**+0.319** [+0.230, +0.409]*	−0.129* (0.0002)	**+0.288** [+0.096, +0.474]*
`gpt-4.1`	**+0.471** [+0.235, +0.706]*	−0.164* (0.0016)	**+0.438** [+0.062, +0.750]*
`qwen-plus`	**+0.191** [+0.043, +0.346]*	−0.059* (0.0092)	+0.142 [−0.062, +0.346]
`gpt-5-mini`	**+0.391** [+0.205, +0.564]*	−0.140* (0.0002)	**+0.380** [+0.153, +0.587]*
`doubao-seed-2.0-lite`	**+0.219** [+0.018, +0.430]*	−0.098* (0.038)	+0.281 [−0.044, +0.632]
`Qwen2.5-14B`	+0.026 [−0.158, +0.211]	0.000 (0.46)	+0.032 [−0.290, +0.355]
`Qwen3-14B`	−0.014 [−0.122, +0.099]	+0.024 (0.79)	−0.133 [−0.305, +0.029]
`Qwen3-30B-A3B`	**+0.126** [+0.015, +0.237]*	−0.050 (0.10)	−0.006 [−0.154, +0.130]
`Granite-3.3-8B`	−0.167 [−0.363, +0.020]	+0.045 (0.85)	−0.135 [−0.354, +0.094]
pooled, API-served (6)	**+0.287** [+0.217, +0.358]*		**+0.294** [+0.210, +0.377]*
pooled, local (4)	+0.007 [−0.106, +0.120]		−0.066 [−0.159, +0.027]
pooled, all ten	**+0.176** [+0.071, +0.281]*		**+0.148** [+0.015, +0.281]*
heterogeneity, all	$Q = 48.47$ (9 df), $I^2 = 0.81$		$Q = 37.20$ (9 df), $I^2 = 0.76$

A working model. A two-state description organises the predictions; no test below depends on it. After contamination the agent's plan is *open* (the false premise not yet folded into a committed course of action, in the agent or in what it has told the customer) or *sealed*. A live turn in which acting on the premise draws no contradiction seals the plan with probability σ; a turn in which the environment rejects the attempt re-opens it. A correction restores success only while the plan is open, so with recoverable loss R_0 its expected value at lag j is $R_0(1-\sigma)^j$ when nothing contradicts the agent, and we take an unprompted re-read to occur only while the plan is open. Hence **P1**: the value of the same correction declines in j; **P2**: an absent record has no premise to seal, so its late supply loses less than a late correction; **P3**: the rung whose false value draws no contradiction seals faster, so re-reading decays faster there and a correction loses more of its value by a given lag ($1-(1-\sigma)^j$ rises with σ; the j=1 minus j=4 difference does so below $\sigma \approx 0.37$, and the pooled shares correspond to $\sigma \approx 0.28$, per family −0.03 to 0.48; Appendix A.6); **P4**: most of the value is gone within a few turns; **P5**: if the seal is in the interaction, a different customer changes the window; if it is in the agent's own context, editing that context late reopens it. §3.1–3.6 test them.

3 RESULTS

3.1 THE SAME CORRECTION, APPLIED LATER, BUYS BACK LESS

Figure 1 shows the curves and Table 2 the paired contrasts. In all six API-served families, correcting at $j = 1$ restores more task success than correcting at $j = 4$ (the contrast we call the window), and the within-unit trend is negative and significant in the same six; after a Holm correction for ten families, four windows and four trends remain. Pooled over the six, the window is $+0.287$ $[+0.217, +0.358]$ ($I^2 = 0.29$; Hartung–Knapp $[+0.193, +0.382]$), and leaving any one out leaves at least $+0.273$ $[+0.199, +0.346]$. Over all ten it is $+0.176$ $[+0.071, +0.281]$, $Q = 48.47$ on 9 df ($I^2 = 0.81$; Hartung–Knapp $[+0.043, +0.309]$), with a 95% prediction interval for a new family of $[-0.188, +0.540]$. In units of the damage the fault does, on the units all four cells share and over the nine families it damages, the correction buys back 0.678 $[0.602, 0.754]$ of it at $j = 1$ and 0.254 $[0.132, 0.376]$ at $j = 4$, a within-unit decline of $+\mathbf{0.420}$ $[+0.317, +0.524]^*$; in the API-served families, 0.690 $[0.609, 0.771]$ and 0.218 $[0.076, 0.360]$. A correction at $j = 1$ is not shown to be complete: in nine families its residual against the no-fault control is -0.07 to -0.30, with the 90% interval below zero in six, and two one-sided tests at $\delta = 0.20$ establish equivalence only on `Qwen3-30B-A3B` (Appendix B).

The locally served families. The split, which we did not anticipate, is most of the heterogeneity of the ten-family pool; over the four local families the window is $+0.007$ $[-0.106, +0.120]$. `Granite-3.3-8B` loses only -0.059 $[-0.235, +0.137]$ to the fault, so it has little to recover. On the two 14B `Qwen` models a correction recovers about half of the damage at either lag (shares 0.47/0.41 and 0.45/0.50, $\sigma \approx 0$ in the working model), and `Qwen3-14B` re-served from its 8-bit checkpoint on the same units and seeds is still damaged ($-\mathbf{0.207}$ $[-0.378, -0.036]^*$) and still has no window (-0.083 $[-0.222, +0.056]$; Appendix A.5). The largest, `Qwen3-30B-A3B`, has a raw window, $+\mathbf{0.126}$ $[+0.015, +0.237]^*$, but none net of the edit (-0.006 $[-0.154, +0.130]$): on the 54 units with all four cells, the edit on a clean prefix loses as much between $j = 1$ and $j = 4$ ($0.716 \to 0.586$) as the faulted arm ($0.642 \to 0.519$); its recovered share, not net of the edit, falls from 0.82 to 0.28. Four families are too few to tell whether scale, open weights or the serving stack separates the groups.

It is not the context surgery. Every window arm deletes a message mid-episode and, for correction, appends a tool result the agent did not ask for; the surgery control performs the identical edits at the identical turns on a clean prefix, on every seed the correction cells carry (one exception, Appendix A.4); against the no-fault control the correction's surgery control ranges from -0.143 to $+0.135$ across arms and lags, none excluding zero, while deletion alone (the withdrawal's) costs up to 0.250 and excludes zero on two families at $j = 1$. Subtracting each faulted arm's own surgery contrast (assuming the edit costs the same on a faulted and a clean trajectory) gives $+0.294$ $[+0.210, +0.377]$ over the API-served families ($I^2 = 0.00$, which reflects the wider intervals differencing brings; Hartung–Knapp $[+0.184, +0.403]$), at least $+0.278$ $[+0.187, +0.369]$ with any one left out, and $+\mathbf{0.148}$ $[+0.015, +0.281]^*$ over all ten, which Hartung–Knapp does not keep ($[-0.005, +0.302]$); it is individually significant in four families. The step budget does not explain it either; in the API-served families 0.00–0.10 of episodes corrected at $j = 1$ and 0.06–0.16 at $j = 4$ exhaust their 20 live steps, at most 0.08 apart. An append-only correction, which keeps the false record (the only option for a harness that cannot rewrite the transcript), has a smaller window ($+0.146$ $[+0.038, +0.255]$, five families) and costs $-\mathbf{0.171}$ $[-0.245, -0.098]^*$ against deletion at $j = 1$ (Appendix A.11).

Writes before the correction. The mechanical alternative to a sealed plan is a write no true record can undo. On the blocking rung, 0.18–0.64 of episodes corrected at $j = 4$ had a write the environment accepted before the correction, and at most 0.02 at $j = 1$. On units with no write before either correction the window pools to $+\mathbf{0.295}$ $[+0.156, +0.434]^*$ over the API-served families (Appendix A.6), though it vanishes on `qwen-plus`; the split is post-treatment, so it describes rather than identifies (§3.6 resets the database instead).

3.2 THE FALSE PREMISE, NOT ONLY THE LATENESS

Gulati et al. (2026) inject ground-truth clarifications at 10–90% of a long-horizon trajectory and find that later ones help less, which could explain Figure 1. The absent-record arms separate lateness from falsity (Table 11, Appendix A.9; curves in Figure 4, Appendix A.10). In T11 the record is missing from the prefix and every attempt to read it before turn j returns an error; at turn j the same true record arrives through the same synthetic tool return as in the correction arm, on the same units and seeds. The arms differ in whether a false version was believed in the meantime, and also in three ways that are not falsity: the withheld arm receives error messages, its prefix has a related read in place of the target, and nothing is deleted at turn j. By $j = 4$ the agent has tried to read the missing record in 0.42–0.95 of episodes, and 0.00–0.10 of withheld episodes end before the record arrives, against 0.07–0.27 of corrected ones. Over the five API-served families where it was run, the withheld window pools to $+\mathbf{0.129}$ $[+0.071, +0.187]^*$, the false premise's window on the same units is $+\mathbf{0.309}$ $[+0.220, +0.398]^*$, and the excess is $+\mathbf{0.180}$ $[+0.082, +0.279]^*$ ($I^2 = 0.00$; Hartung–Knapp $[+0.041, +0.320]$); counting every scheduled episode, so that attrition selects nothing, it is $+\mathbf{0.192}$ $[+0.077, +0.307]^*$ (Hartung–Knapp $[+0.026, +0.358]$). If the error messages hurt no less by turn four than by turn one, the withheld window overstates the cost of lateness, and the excess understates what the false belief costs; taking the correction's own edit out with its surgery control, which the withheld arm has no counterpart to, raises the excess to $+\mathbf{0.202}$ $[+0.099, +0.305]^*$ (Hartung–Knapp $[+0.056, +0.348]$). Adding `Qwen3-14B` (own excess -0.083 $[-0.278, +0.125]$) gives $+\mathbf{0.143}$ $[+0.016, +0.270]^*$, which Hartung–Knapp does not keep. At $j = 1$ the two arms differ by -0.043 $[-0.108, +0.022]$; at $j = 4$ corrected minus withheld is $-\mathbf{0.281}$ $[-0.459, -0.103]^*$ ($I^2 = 0.76$; pooled differences need not add up). This is what P2 predicts, that the recoverability of a false belief decays beyond lateness; Laban et al. (2026) document a related failure in conversation, where an early wrong assumption resists later information.

3.3 AFTER A WITHDRAWAL, THE AGENT STOPS LOOKING

Withdrawal leaves a hole where the record was, and recovery goes with whether the agent goes back for it (Table 9, Appendix A.7: residual against the no-fault control and re-read probability per lag). Deleting the premise before the first live turn costs significantly on three families; deleting it after four turns costs every family the fault damages at least 0.33 except `Qwen3-30B-A3B` (0.17, not significant; residuals include the deletion's own cost, §3.1). Re-read probability falls with j in ten of ten families, monotonically in seven (`Qwen2.5-14B`, `doubao-seed-2.0-lite` and `Granite-3.3-8B` tick up once), mostly between $j = 0$ and $j = 1$ in five, and across the 40 policy-by-lag cells mean task success is 0.56 given a re-read and 0.10 without (a descriptive, post-treatment split). In §3.2, by contrast, an agent that lacks a record tries to fetch it; one that has acted on a false version stops looking. The two 14B `Qwen` models, with no correction window, show the second too: re-read probability falls from 0.48 to 0.08 and from 0.81 to 0.41.

3.4 TELLING THE AGENT IT WAS WRONG: NO RELIABLE BENEFIT

Retraction adds to silent correction what it leaves out: that the earlier record was false and the agent's own statements suspect. Two placebos send a same-channel message without it, one restating the policy's rule to confirm before any change and one with no instruction at all (Appendix A.3, with Table 5). At $j = 4$ the retraction adds $+0.036$ $[-0.064, +0.136]$ to silent correction over seven families ($I^2 = 0.64$; Hartung–Knapp $[-0.100, +0.172]$), significantly on `gpt-4o-mini` and `gpt-4.1` and negative without significance on three. Against the instruction-free placebo, run at $j = 4$ on five families, it adds $+\mathbf{0.174}$ $[+0.104, +0.245]^*$ (positive in five of five; $I^2 = 0.00$; Hartung–Knapp $[+0.075, +0.274]$), partly because on `deepseek-v4-flash` the empty message itself costs $-\mathbf{0.256}$ $[-0.462, -0.038]^*$; against the policy-restating placebo it adds -0.001 $[-0.065, +0.064]$, inside ± 0.10 (90% Hartung–Knapp interval $[-0.076, +0.075]$). Late, a message that tells the agent to confirm before acting does as well as the retraction, and only on `gpt-4.1` does the retraction's content add to it. At $j = 1$ the empty placebo costs $-\mathbf{0.075}$ $[-0.123, -0.027]^*$ against silent correction ($I^2 = 0.00$; Hartung–Knapp $[-0.138, -0.012]$), negative in five of six families; on the same six the retraction costs $-\mathbf{0.051}$ $[-0.099, -0.002]^*$, which does not survive the Hartung–Knapp correction ($[-0.114, +0.013]$), and retraction minus the empty placebo is $+0.027$ $[-0.023, +0.076]$. At short lag the cost comes from a harness message arriving rather than from its content.

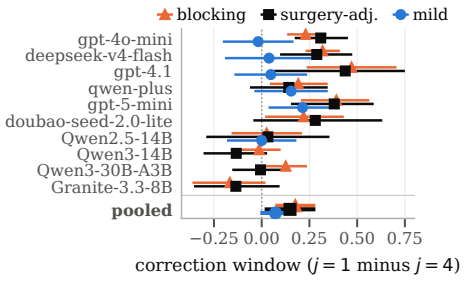


Figure 2: The correction window per policy with 95% intervals: blocking rung raw (▲) and surgery-adjusted (■), each on its full design, and mild rung (•; six families). Pooled: random effects over all ten (mild: six).

Table 3: The ladder: success under the fault minus the no-fault control at $k = 0$, on the order record unless marked. *: interval excludes zero.

policy	inert	mild	blocking
`gpt-4o-mini`	−0.091	−0.357*	−0.369*
`deepseek-v4-flash`	−0.045	−0.385*	−0.811*
`gpt-4.1`	−0.056	−0.619*	−0.810*
`qwen-plus`	+0.042	−0.107	−0.500*
`gpt-5-mini`	−0.083	−0.429*	−0.798*
`doubao-seed-2.0-lite`	−0.150	−0.348*	−0.609*
`Qwen2.5-14B`	0.000	−0.250*	−0.433*
`Qwen3-14B`	−0.091	−0.108	−0.261*
`Qwen3-30B-A3B`	+0.077	−0.065	−0.230*
`Granite-3.3-8B`	−0.167	+0.118	−0.059
`gpt-4o-mini`, product	−0.036	−0.143	−0.321*
`gpt-5-mini`, product	−0.222	−0.667*	−0.833*

3.5 What the false value licenses

The ladder. Table 3 gives the fault effect at $k = 0$ on each rung, for ten order-record and two product-record arms (§3.7). The effect is monotone in eleven of twelve arms by point estimate; the exception is `Granite-3.3-8B`, which the fault barely damages on any rung. No inert-rung interval excludes zero, and the ten order-record inert effects pool to -0.050 $[-0.114, +0.013]$, equivalent to zero within ± 0.20 but not within ± 0.10: an edit to a field the task never uses costs little, and the mild and blocking rungs cost much more. In the row used below, `gpt-4o-mini`'s mild and blocking rungs land together, at -0.357 and -0.369, while every other order-record arm the fault damages separates them by 0.15 to 0.43.

Re-reading decays faster where nothing contradicts the agent, in some families. The probability of re-reading the corrupted record after a withdrawal (Appendix A.8, Figure 3) falls in j on the blocking rung, steeply in three families; on the mild rung, where the agent's attempts are rejected, it moves less. The paired within-unit difference in slopes is negative and significant in three of six families (`deepseek-v4-flash`, `gpt-4.1`, `gpt-5-mini`), negative without significance on `qwen-plus`, and absent on `gpt-4o-mini` and `Qwen2.5-14B`; it pools to $\mathbf{-0.071}$ $[-0.134, -0.009]^*$ ($I^2 = 0.78$), and to $\mathbf{-0.090}$ $[-0.152, -0.028]^*$ over the five families whose mild rung was milder, neither under Hartung–Knapp.

The window is wider on the blocking rung. P3 predicts a wider window on the blocking rung, for sealing rates below $\sigma \approx 0.37$. Pairing each unit's own window under the two rungs (per-family gaps in Table 10, per-rung windows in Figure 2), the window is wider on the blocking rung in six of six families, significant in two, and pools to $+\mathbf{0.206}$ $[+0.097, +0.314]^*$ with $Q = 4.01$ on 5 df (Hartung–Knapp $[+0.063, +0.348]$). Run as a full design, the mild rung pools to $+0.072$ $[-0.007, +0.152]$, one family of six significant, against $+0.264$ $[+0.166, +0.363]$ on the blocking rung of the same six families. By point estimate four families show both a re-read gap and a window gap, and `gpt-4o-mini` and `Qwen2.5-14B` a window gap without a re-read gap. `gpt-4o-mini`'s rungs do equal damage and its re-read slopes are alike, yet its window gap is $+\mathbf{0.252}$ $[+0.069, +0.447]^*$; without it the gap pools to $+\mathbf{0.183}$ $[+0.050, +0.316]^*$, positive in five of five but not under Hartung–Knapp ($[-0.005, +0.371]$). So the window is wider on the blocking rung, and reduced re-reading goes with it in four families but does not account for all of it; six families leave open whether what the value licenses or the damage it does sets the width.

3.6 Where the loss lives

If what closes the window lives in the interaction, changing the customer should change it (P5). We reran the control, the fault, the corrections at $j \in \{1, 4\}$ and their surgery controls at seeds 1–3 with `deepseek-v4-flash` as the customer of `gpt-4o-mini` and `gpt-5-mini`, and `gpt-4.1-mini` as that of `gpt-4o-mini` (Appendix A.5). The fault does comparable damage

under every customer. Among episodes that reach the late correction the second customer's window is smaller, by $+0.182$ $[+0.095, +0.269]$ over the three pairs (Hartung–Knapp $[-0.009, +0.373]$), but the customer also changes which episodes reach it. For `gpt-4o-mini` the survivor window vanishes under both second customers. Keeping every scheduled episode at its observed outcome (one that ended before turn four counts as uncorrected), the second customer's window is positive in all three pairs and differs from the default's by $+0.033$ $[-0.042, +0.109]$ (Hartung–Knapp $[-0.133, +0.200]$). Three pairs from two policies cannot separate a customer-dependent window from customer-dependent attrition.

The converse tests edit what the harness controls at the late correction (Table 14; three families, seeds 1–2), each net of the same edit on a clean prefix, assuming the costs add. Scrubbing the agent's replies buys nothing (-0.060 $[-0.268, +0.147]$). Rewinding its live transcript buys back $+0.161$ $[-0.018, +0.340]$ of a surgery-adjusted window of $+\mathbf{0.281}$ $[+0.157, +0.405]^*$: most of it on `gpt-4o-mini` and `gpt-5-mini`, none on `deepseek-v4-flash`. Resetting the database alone undoes clean episodes' legitimate writes ($-\mathbf{0.599}$ $[-0.742, -0.455]^*$), so it has no usable control. Rewinding and resetting together, the fault's residual damage after a late correction goes from $-\mathbf{0.245}$ $[-0.382, -0.108]^*$ to $+0.000$ $[-0.127, +0.127]$ on `gpt-4o-mini`, but only because the edit costs clean episodes as much ($0.686 \to 0.392$): faulted success does not rise ($0.441 \to 0.392$); on `deepseek-v4-flash` it stays at $-\mathbf{0.404}$ $[-0.558, -0.250]^*$, and only returning the customer simulator to its opening message as well takes it to -0.068 $[-0.136, +0.000]$, against $-\mathbf{0.455}$ $[-0.659, -0.250]^*$ after the plain late correction on the same units. There faulted success rises from 0.318 to 0.818, the one late edit that restores it; no edit to the agent's side alone reliably does.

3.7 A SECOND FAULT FAMILY

The product catalogue is a different record shape read by a different tool, and retail policy makes one field decisive: an exchange may only move to an `available` variant. The rungs are built the same way (blocking: `available` set to false on the target variant; mild: the target variant's options swapped with another variant's, which need not draw a rejection; inert: an untouched variant's price). On two OpenAI policies at seed 1 the fault effect (`gpt-4o-mini` $0.536 \to 0.214$, `gpt-5-mini` $0.833 \to 0.000$) and the ladder's order replicate; the correction window is $+0.115$ $[-0.115, +0.346]$ ($n = 26$) and $+\mathbf{0.222}$ $[+0.056, +0.444]^*$ ($n = 18$), success withdrawn at $j = 0$ minus at $j = 4$ is $+\mathbf{0.407}$ $[+0.185, +0.630]^*$ and $+\mathbf{0.588}$ $[+0.353, +0.824]^*$, and re-reading decays on both. Two policies of one vendor make this a replication of direction, not a pooled estimate.

3.8 ROBUSTNESS OF THE FAULT EFFECT

In the five position sweeps ($k \in \{0, 1, 2, 4, 7\}$, seed 1: `gpt-4o-mini` on the blocking rung, `deepseek-v4-flash` and `Qwen2.5-14B` on both), every arrangement-matched fault effect is significant: one corrupted field in one already-read observation removes 0.23 to 0.85 of task success, in absolute points. The fault effect also survives every customer swap (§3.6 and below). `Qwen2.5-14B`'s default customer is the same local model. With API-served `gpt-4o-mini` as its customer, its mild-rung window stays flat ($+0.000$ $[-0.160, +0.160]$); on the blocking rung (seeds 1–3) all 52 units rather than 38 reach the late correction and the window is $+0.087$ $[-0.019, +0.199]$, against $+0.026$ $[-0.158, +0.211]$ with the local customer; `Qwen3-14B`, served with the API customer throughout, has no window either (§3.1).

4 RELATED WORK

Timing and severity. Gulati et al. (2026) (§3.2) is the closest prior result; they find a recovery point only for outcome-critical variants, so a severity-by-timing interaction is on record. What is new is a false rather than absent payload, a severity ladder whose rungs differ in environment feedback, and per-unit paired windows with a surgery control. Error cascades among agents (Xie et al., 2026; Itkin, 2026) and feedback over turns (Wang et al., 2024) are adjacent.

Silent tool failure. Advani (2026) finds agents claiming success at 45–48% of failures in single-control τ^2-bench domains (Barres et al., 2026). Panthi & Abdelfattah (2026), also on τ-bench retail,

vary when a warning arrives, not when the true record does; Xu et al. (2026) corrupt the same `status` field and score next-action log-probabilities without a correction; Zhang et al. (2026) study conflicting early and late evidence without a lag sweep; Yang et al. (2026) find no intervention level that reliably mitigates tool over-trust; injected tool output (Debenedetti et al., 2024) is the hostile version of our benign fault.

Self-correction and belief. Self-correction gains come from external signal (Huang et al., 2024; Stechly et al., 2025), which tools can supply (Gou et al., 2024); a located error is easier to fix (Tyen et al., 2024), yet our retraction names the record and has no reliable effect, and a committed early error constrains what a model generates next (Zhang et al., 2024). Others characterise when models flag, fix or abandon positions (Chen et al., 2026a; Yang & Jia, 2025; Chen et al., 2026c; Sharma et al., 2024) and why restated bad content stays salient (Gumaan, 2026); Wang et al. (2026a), the nearest prior art for the re-read result, show embodied agents overlooking contradicting feedback. Mehta & Datta (2026) and Chen et al. (2026b) study what persists in agent context; Yao et al. (2026), Yu et al. (2026), Zhuang et al. (2026) and Shen (2026) roll back or evict execution or context state, where our resets also reach the database and the customer. A false record followed by a true one is a conflict within the context (Xu et al., 2024), where most work pits context against parametric knowledge (Longpre et al., 2021; Xie et al., 2024; Wu et al., 2024). The continued-influence effect in humans (Johnson & Seifert, 1994; Lewandowsky et al., 2012; Ecker et al., 2022) is the obvious analogue: a retraction helps little unless an alternative account replaces what it removes, as the true record does here.

5 Limitations

One environment, one default customer. τ-bench airline was ported, but four policies solve 0–3 of its 14 eligible tasks at the clean screen (Appendix B); every API arm, the `gpt-4o-mini` policy's included, uses the `gpt-4o-mini` customer, and only three policies were rerun with others (§3.6, §3.8). **Small arms**: 17–99 task units per arm. **Families are not independent**: three are OpenAI's and four Alibaba's, nine share one customer and most draw largely on one pool of 55 tasks (Appendix A.4 gives the window on train-split units), and leaving a family out cannot reveal a bias they share. **Survivors**: 0–27% of units never reach the $j = 4$ correction and up to 39% the $j = 4$ withdrawal; the intention-to-treat window is $+\mathbf{0.326}$ $[+0.224, +0.428]^*$ over the API-served families and $+\mathbf{0.200}$ $[+0.086, +0.314]^*$ over all ten (Appendix A.6). **Seeds were added after a null** on `gpt-4o-mini`; the first seed of every arm alone pools to $+\mathbf{0.163}$ $[+0.044, +0.282]^*$ over all ten. **One field, one position**: the headline is the blocking rung at $k = 0$, the most recent read, where lag since acting is not separated from time since reading (Appendix B); the adjusted window assumes the edit costs the same with and without the fault. **Partial designs**: the late edits cover three families (the rewind with the reset two, the full reset one), the customer swap two policies, the withheld record, append-only arm and empty placebo at $j = 4$ five or six, the product family two. **Controls not run**: a customer reset on more than one family, a database reset that keeps legitimate writes, a surgery control for the append-only arm, a withheld arm without error messages, and a customer-directed retraction. **The correction is an oracle's**, not a detector's; **the retraction is one wording, channel and placement**. **Net of the edit, no window is detected in the four open-weight families**, and in them open weights, scale (8 to 30B, at most 14B active) and local serving go together; quantisation is varied on one, scale only as far as one 24 GB GPU serves.

6 Conclusion

In the API-served agents we studied, a false observation stays fixable only briefly: the same correction buys back about two-thirds of the loss one turn after the false read and a fifth four turns after. Lateness is part of that decline and falsity a further part; neither a late edit to the agent's side nor telling it that it was wrong reliably undoes the loss. A harness should correct early, budget detection latency in agent actions, and catch first the faults that license a refusal. The open-weight families we served locally show no detectable decline net of the edit's own cost; larger ones, other environments and human users are the next tests.

REPRODUCIBILITY STATEMENT

Every estimate in the main text and Appendix A, except the position sweeps of §3.8, is regenerated from the raw episode files by `code/refresh_all.sh` without calling any model; the shipped result files are its output, and a clean run reproduces them byte-identically under different `PYTHONHASHSEED` values. The position sweeps and Appendix B's calibration numbers are outputs of the archive's earlier analysis scripts (`code/analyze_formal.py`, `code/calibrate_judge.py`), shipped as produced. The anonymised supplementary archive contains the instrument, the analysis code, all 30,942 episodes with full step traces, and the offline validator. Re-running episodes needs API keys for the four API vendors and a GPU for the four local families; requests set temperature 0 (Appendix A.2 gives the exceptions), but served models are not deterministic. A *seed* is a repeat of the episode with that value passed as the policy call's seed; whether an endpoint honours it is not verified, so seeds are replicates. Appendix A lists the conditions behind every main-text result, and Appendix B the negative results and withdrawn claims.

ETHICS STATEMENT

The experiments use a public benchmark with a simulated customer and no human subjects. The faults studied are the kind that occur in tool pipelines, and the work is meant to inform correction and monitoring in deployed agents. The same results say which false values do the most lasting damage, which could help someone able to tamper with a tool response; they also suggest a defence: supplying the true record within a turn of the false read.

AI USE STATEMENT

In this work, we used generative AI tools to assist in writing, editing, and debugging code, including code used to implement our methods and experiments. We did not use generative AI tools for any of the other tasks that require disclosure: in particular, we did not use them to generate synthetic data (the episodes we analyse are, by design, produced by the language models under study interacting with the benchmark's model-based simulated customer; these models are the object of the experiments, not tools that assisted the research), to develop theoretical models or conceptual frameworks, to formulate mathematical claims or assist in proofs, to propose or refine hypotheses, to design the research methodology or experiments, to assist with translation, to clean or reformat datasets, or to support data analysis or the interpretation of results. Additionally, we used generative AI tools to edit the manuscript to improve readability and clarity, and to obtain simulated peer-review-style feedback on drafts, which informed revisions to the writing and presentation. All AI-assisted code was reviewed and tested for correctness by the authors, and all AI-suggested edits and feedback were reviewed and adopted at the authors' discretion; the scientific content, claims, and results are entirely the authors' own. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Laksh Advani. From confident closing to silent failure: Characterizing false success in LLM agents. *arXiv preprint arXiv:2606.09863*, 2026.

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2-bench: Evaluating conversational agents in a dual-control environment. In *International Conference on Machine Learning*, 2026.

Kuan-Yen Chen, Fang-Yi Su, Shih-Yen Lin, Bao Li, and Jung-Hsien Chiang. The self-correction illusion: Role relabeling gates explicit error flagging in large language models. *arXiv preprint arXiv:2606.05976*, 2026a.

Yixiong Chen, Xinyi Bai, and Alan Yuille. The compliance trap: Diagnosing how AI agents consume conflicting memory. *arXiv preprint arXiv:2607.10608*, 2026b.

Zixuan Chen, Hao Lin, Zizhe Chen, Yizhou Tian, Garry Yang, Depeng Wang, Ya Guo, Huijia Zhu, and James Cheng. Knowing but not correcting: Routine task requests suppress factual correction in LLMs. *arXiv preprint arXiv:2605.05957*, 2026c.

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 (Datasets and Benchmarks Track)*, 2024.

Rebecca DerSimonian and Nan Laird. Meta-analysis in clinical trials. *Controlled Clinical Trials*, 7 (3):177–188, 1986.

Ullrich K. H. Ecker, Stephan Lewandowsky, John Cook, Philipp Schmid, Lisa K. Fazio, Nadia Brashier, Panayiota Kendeou, Emily K. Vraga, and Michelle A. Amazeen. The psychological drivers of misinformation belief and its resistance to correction. *Nature Reviews Psychology*, 1: 13–29, 2022.

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 (ICLR)*, 2024.

Anmol Gulati, Hariom Gupta, Elias Lumer, Sahil Sen, and Vamse Kumar Subbiah. Ask early, ask late, ask right: When does clarification timing matter for long-horizon agents? *arXiv preprint arXiv:2605.07937*, 2026.

Esmail Gumaan. Feedback that backfires: Why small language model agents repeat the call they just watched fail. *arXiv preprint arXiv:2608.23651*, 2026.

Joachim Hartung and Guido Knapp. A refined method for the meta-analysis of controlled clinical trials with binary outcome. *Statistics in Medicine*, 20(24):3875–3889, 2001.

Julian P. T. Higgins and Simon G. Thompson. Quantifying heterogeneity in a meta-analysis. *Statistics in Medicine*, 21(11):1539–1558, 2002.

Julian P. T. Higgins, Simon G. Thompson, and David J. Spiegelhalter. A re-evaluation of random-effects meta-analysis. *Journal of the Royal Statistical Society: Series A*, 172(1):137–159, 2009.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. In *International Conference on Learning Representations*, 2024.

Igor Itkin. Delayed verification destabilizes multi-agent LLM belief: Instability thresholds and optimal corrector placement. *arXiv preprint arXiv:2606.27409*, 2026.

Hollyn M. Johnson and Colleen M. Seifert. Sources of the continued influence effect: When misinformation in memory affects later inferences. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 20(6):1420–1436, 1994.

Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. LLMs get lost in multi-turn conversation. In *International Conference on Learning Representations*, 2026.

Stephan Lewandowsky, Ullrich K. H. Ecker, Colleen M. Seifert, Norbert Schwarz, and John Cook. Misinformation and its correction: Continued influence and successful debiasing. *Psychological Science in the Public Interest*, 13(3):106–131, 2012.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024.

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. Entity-based knowledge conflicts in question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021.

Aman Mehta and Anupam Datta. Plans don't persist: Why context management is load bearing for LLM agents. *arXiv preprint arXiv:2606.22953*, 2026.

Evan Miller. Adding error bars to evals: A statistical approach to language model evaluations. *arXiv preprint arXiv:2411.00640*, 2024.

Sugam Panthi and Rabab Abdelfattah. Outcome monitors: Recovery affordances for silent tool failures. *arXiv preprint arXiv:2608.19303*, 2026.

Christian Röver, Guido Knapp, and Tim Friede. Hartung-Knapp-Sidik-Jonkman approach and its modification for random-effects meta-analysis with few studies. *BMC Medical Research Methodology*, 15:99, 2015.

Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R. Bowman, Newton Cheng, Esin Durmus, Zac Hatfield-Dodds, Scott R. Johnston, Shauna Kravec, Timothy Maxwell, Sam McCandlish, Kamal Ndousse, Oliver Rausch, Nicholas Schiefer, Da Yan, Miranda Zhang, and Ethan Perez. Towards understanding sycophancy in language models. In *International Conference on Learning Representations*, 2024.

Chen Shen. What eviction destroys: A restore-counterfactual audit of forgetting in agent memory. *arXiv preprint arXiv:2609.08279*, 2026.

Kaya Stechly, Karthik Valmeekam, and Subbarao Kambhampati. On the self-verification limitations of large language models on reasoning and planning tasks. In *International Conference on Learning Representations*, 2025.

Gladys Tyen, Hassan Mansoor, Victor Cărbune, Peter Chen, and Tony Mak. LLMs cannot find reasoning errors, but can correct them given the error location. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024.

Hanlin Wang, Chak Tou Leong, Jian Wang, and Wenjie Li. Seeing isn't believing: Mitigating belief inertia via active intervention in embodied agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 37796–37815, 2026a.

Ruipeng Wang, Yuxin Chen, Yukai Wang, Chang Wu, Junfeng Fang, Xiaodong Cai, Qi Gu, Hui Su, An Zhang, Xiang Wang, Xunliang Cai, and Tat-Seng Chua. AgentNoiseBench: Benchmarking robustness of tool-using LLM agents under noisy condition. *arXiv preprint arXiv:2602.11348*, 2026b.

Xingyao Wang, Zihan Wang, Jiateng Liu, Yangyi Chen, Lifan Yuan, Hao Peng, and Heng Ji. MINT: Evaluating LLMs in multi-turn interaction with tools and language feedback. In *International Conference on Learning Representations (ICLR)*, 2024.

Kevin Wu, Eric Wu, and James Zou. ClashEval: Quantifying the tug-of-war between an LLM's internal prior and external evidence. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, 2024.

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts. In *International Conference on Learning Representations*, 2024.

Yizhe Xie, Congcong Zhu, Xinyue Zhang, Tianqing Zhu, Dayong Ye, Minfeng Qi, Huajie Chen, and Wanlei Zhou. From spark to fire: Modeling and mitigating error cascades in LLM-based multi-agent collaboration. *arXiv preprint arXiv:2603.04474*, 2026.

Jiachen Xu, Torben Bach Pedersen, Zhongming Yao, Xiaoyu Zhang, and Yushuai Li. Robustness analysis of agentic AI to inconsistent and incomplete tool responses. *arXiv preprint arXiv:2608.22676*, 2026.

Rongwu Xu, Zehan Qi, Zhijiang Guo, Cunxiang Wang, Hongru Wang, Yue Zhang, and Wei Xu. Knowledge conflicts for LLMs: A survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024.

Hoyeol Yang, Woojung Song, Taewon Kim, Jonghyun Song, Seoyeon Park, and Yohan Jo. Agents trust tools too much: Measuring reliance on unreliable tools. *arXiv preprint arXiv:2609.05587*, 2026.

Yuqing Yang and Robin Jia. When do LLMs admit their mistakes? Understanding the role of model belief in retraction. *arXiv preprint arXiv:2505.16170*, 2025.

Chao Yao, Yangbo Wei, Zhen Huang, Junhong Qian, Chenle Chen, Shaoqiang Lu, Chen Wu, and Lei He. Forgetting without restarting: Execution-state unlearning for stateful LLM agents. *arXiv preprint arXiv:2609.04875*, 2026.

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 (ICLR)*, 2025.

Caili Yu, Yiqi Wang, Jiaqi Zhang, Yiqun Duan, Mingkai Zheng, Zhangkai Wu, Kaize Shi, and Taotao Cai. From faulty memories to corrected actions: Dependency-guided rollback repair for memory-augmented agents. *arXiv preprint arXiv:2608.10502*, 2026.

Chubin Zhang, Zhenglin Wan, Xingrui Yu, Pengfei Zhou, Wangbo Zhao, Jingxuan Wu, Yaxin Zhou, and Ivor Tsang. Don't blindly trust it: How unreliable feedback breaks tool-using LLM agents. *arXiv preprint arXiv:2606.21409*, 2026.

Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A. Smith. How language model hallucinations can snowball. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Yu Zhuang, Kefei Chen, Yitong Duan, Shuxin Zheng, Jian Li, and Xu-Yao Zhang. AgentRewind: Recoverable execution for long-horizon LLM agents. *arXiv preprint arXiv:2608.14380*, 2026.

A Design

A.1 Prefix construction and fault injection

For each task the ground-truth action list is parsed for the entities it touches. A nine-step read-only prefix is assembled from the user record, up to three orders the ground truth touches and up to four of their products, padded to nine with the user's other orders, an e-mail lookup of the user, the product-type list and further catalogue products, executed against the live environment, and replayed to the agent as its own past assistant tool calls and tool results after the system prompt and the customer's opening message. Read-only calls leave the database unchanged, so the list can be permuted; *arrangement* k places the target observation k reads from the end. A permutation can place a read before the lookup that produced its identifier; that is common to every cell at a given k, including its no-fault control, and the window arms all use $k = 0$. Faults are written into the replayed observation text after execution; `env.step` is never wrapped, because τ-bench's `calculate_reward` replays the ground-truth actions through `step` at scoring time and a wrapper would fire again and corrupt the state hash. The offline validator replays the benchmark's own ground-truth solutions through the instrumented environment and scores $60/60$ (30 retail and 30 airline ground-truth replays) at reward 1.000; separate offline tests drive scripted episodes through the withholding, rewind and scrub conditions and check the transcript the agent sees.

For the *absent* arms (T10, T11), the prefix is rebuilt with one more read and the target read dropped, so it keeps nine read-only steps; the runner asserts that the dropped step is the one that would have carried the fault. In T11 the runner answers the agent's live reads of the target before turn j itself, with the error text of §2, without calling `env.step`; reads are read-only, so the database and the reward are unaffected.

A.2 Conditions

Table 4 lists the conditions behind every main-text result; tiers 5–8 and 10–13 are the T-labels of §2, and tiers 0, 3 and 9 group the unintervened cells, the withdrawal before the first live turn and the exploratory $j = 8$ lag. Each run is keyed on (policy, environment, split, fault variant, condition, task, seed), so an interrupted run resumes without duplicating episodes; the customer swaps write to their own files. The live budget is 20 agent steps in every condition listed; j counts live agent steps, each a tool call or a message to the customer. A tighter per-task budget (the clean live-step count plus three) exists only in the age–budget arms of Appendix B.

Decoding. Every policy request sets temperature 0 with a 900-token completion ceiling, with the tool schema passed and tool choice left automatic. `gpt-5-mini` gets a 4,000-token ceiling, its reasoning left on at the provider's default effort, and may not honour the temperature setting. `deepseek-v4-flash` is called with thinking disabled, as policy and as customer. A failed request is retried up to six times; an episode whose request still fails is not recorded and is re-run when the arm resumes. An unparseable reply is retried once with a message asking for the required form, and if still unparseable is sent to the customer as a message. The customer is called at temperature 0 with a 300-token ceiling. `Qwen2.5-14B-Instruct-AWQ` is served by vLLM with a 20,480-token context window; `Granite-3.3-8B-Instruct`, `Qwen3-14B-AWQ`, `Qwen3-30B-A3B-GPTQ-Int4` and the official FP8 checkpoint of `Qwen3-14B` (the 8-bit reserving of §3.1) with a 32,768-token window and a key–value cache stored in 8 bits, the `Qwen3` models with thinking switched off through their chat template, and `Granite-3.3-8B` with a chat template that renders the agent's earlier tool calls (its stock template drops them). For these, a list-typed tool argument sent as a JSON-encoded string is decoded before the call; the API arms never send one, and `Qwen2.5-14B` did in 3 of 3,973 episodes, which were not decoded.

A.3 The notices

The retraction (T7) appends the following as a system-channel message immediately after the corrected record, with the entity name filled in:

> Data-integrity notice from the harness: one earlier tool observation in this conversation contained an incorrect value and has been removed. The authoritative record for *[entity]* is the tool result immediately above. Disregard any earlier statement that conflicts with it, including statements you made yourself, and re-check any conclusion you drew from the removed observation before acting.

The first placebo (T7b) appends, on the same channel at the same turn:

> Routine checkpoint from the harness: this conversation is being logged for quality review. No action is required beyond continuing to help the user with their request according to the operating policy, using the tools available to you, and confirming details with the user before making any change.

Its last clause restates a rule the retail policy already imposes, but it is an instruction nonetheless, so a second placebo (T7c) with no instruction at all was run at $j = 1$, the lag at which a message's own cost shows (§3.4), and then at $j = 4$ on five of the seven families with a retraction arm:

> Routine checkpoint from the harness: this conversation is being logged for quality review.

Table 5 reports the second; at $j = 4$ it changes success against silent correction by -0.061 $[-0.135, +0.013]$ pooled (`gpt-4o-mini` $+0.006$, `deepseek-v4-flash` -0.256^{*}, `gpt-4.1` $+0.000$, `gpt-5-mini` -0.042, `Qwen3-14B` -0.161), and on its units the retraction adds $+\mathbf{0.097}$ $[+0.011, +0.182]^{*}$ to silent correction (Hartung–Knapp $[-0.034, +0.228]$). Against the first at $j = 4$ the retraction adds -0.001 $[-0.065, +0.064]$ pooled ($I^2 = 0.38$): `gpt-4o-mini` -0.020 $[-0.180, +0.140]$, `deepseek-v4-flash` -0.095 $[-0.286, +0.095]$, `gpt-4.1` $+\mathbf{0.357}$ $[+0.143, +0.643]^{*}$, `qwen-plus` -0.087 $[-0.348, +0.174]$, `gpt-5-mini` $+0.000$ (identical outcomes on all 13 shared units), `Qwen2.5-14B` $+0.000$ $[-0.143, +0.143]$, `Qwen3-14B` -0.065 $[-0.258, +0.129]$; the placebo itself changes success at $j = 4$ by -0.027 $[-0.090, +0.036]$ against silent correction. Against silent correction at $j = 1$ the first changes success by -0.082 $[-0.180, +0.016]$ pooled (`gpt-4o-mini` $-\mathbf{0.095}$ $[-0.179, -0.012]^{*}$, `deepseek-v4-flash` $-\mathbf{0.256}$ $[-0.410, -0.115]^{*}$, `gpt-4.1` -0.143 $[-0.429, +0.143]$, `qwen-plus` $+0.048$ $[-0.060, +0.155]$, `gpt-5-mini` -0.036 $[-0.155, +0.083]$), and retraction minus the first placebo at $j = 1$ is $+0.023$ $[-0.077, +0.124]$ (retraction minus the second is $+0.027$ over six families, with a narrower interval).

A.4 Task selection and seeds

Candidate tasks are those whose ground truth contains at least one non-`respond` action and references an order (or, for the product family, a product) whose record can carry the fault: 55 in the retail

Table 4: Conditions. The tier is the T-label used in the text. j counts live agent steps before the intervention; $j = 0$ means before the first live turn.

tier	condition	description
0	`clean_retest`	no fault, independent re-run (the baseline)
0	`permclean_k{1,2,4,7}`	no fault, prefix permuted to arrangement k (arrangement control)
0	`f_k{0,1,2,4,7}_bhi`	fault at arrangement k, never intervened on (position sweep; `bhi`: the 20-step budget)
3	`f_k0_bhi_excise`	fault, withdrawn before the first live turn ($j = 0$)
5	`f_k0_bhi_excise_j{1,2,4}`	withdrawal after j live turns
6	`f_k0_bhi_refetch_j{1,2,4}`	silent correction after j live turns
7	`f_k0_bhi_retract_j{1,2,4}`	correction plus the data-integrity notice
7b	`f_k0_bhi_placebo_j{1,4}`	correction plus the policy-restating placebo notice
7c	`f_k0_bhi_placebopure_j{1,4}`	correction plus the instruction-free notice
6	`f_k0_bhi_append_j{1,4}`	append-only correction: true record appended, false one kept
8	`clean_excise_j{1,2,4}`	surgery control: the same deletion on a clean prefix
8	`clean_refetch_j{1,2,4}`	surgery control: the same deletion and insertion on a clean prefix
9	`f_k0_bhi_excise_j8`, `f_k0_bhi_refetch_j8`, `clean_refetch_j8`	the exploratory longer lag
10	`missing_inject_j{1,2,4}`	record absent from the prefix; true record supplied at turn j
11	`missing_withhold_j{1,4}`	as tier 10, and every earlier read of the record returns a transient error
12	`f_k0_bhi_refetch_rewind_j4`	correction at $j = 4$, the agent's live transcript removed
12	`f_k0_bhi_refetch_scrub_j4`	correction at $j = 4$, the agent's own replies replaced by a placeholder
12	`clean_refetch_rewind_j4`, `clean_..._scrub_j4`	the rewind and the scrub on a clean prefix
13	`f_k0_bhi_refetch_restore_j4`	correction at $j = 4$ with the database reset to its state at the end of the prefix, undoing every live write; with `clean_` for `f_k0_bhi_`, on a clean prefix
13	`..._refetch_rewindrestore_j4`	the rewind and the reset together, faulted and clean
13	`..._refetch_fullreset_j4`	the rewind and the reset, and the customer simulator's history cut back to its opening message
—	`customer swap`	tier 0 at $k = 0$ and the $j \in \{1, 4\}$ cells of tiers 6 and 8 (`clean_refetch`), seeds 1–3, run with `--user-model deepseek-v4-flash` (`gpt-4o-mini`, `gpt-5-mini`) or `gpt-4.1-mini` (`gpt-4o-mini`) (§3.6)

Every faulted condition above is run under `--fault-variant cancelled` (blocking), `flip` (mild) or `item` (inert); the ladder (§3.5) is the three variants of `f_k0_bhi`; the mild-rung window is tier 6 under `flip`, and its re-read curve tier 5. The product family (§3.7) uses the same condition names with `prod_block`, `prod_swap` and `prod_inert`. The de-confound arm (§3.8) is the two tier-0 cells at $k = 0$ and the three withdrawal and three correction cells of tiers 5–6 under `flip`, with `--user-model gpt-4o-mini` on the local policy (414 of 8×52 designed episodes), and on the blocking rung the tier-0 cells at $k = 0$ and the $j \in \{1, 4\}$ cells of tiers 6 and 8, seeds 1–3.

test split. Each policy is screened on every candidate at seed 0 with a clean prefix, and the tasks it solves form its task set; `gpt-4o-mini`, `Qwen2.5-14B` and `deepseek-v4-flash` were also screened on 89, 145 and 100 train-split candidates, and `Qwen3-14B` and `Granite-3.3-8B` on the first 60 and `Qwen3-30B-A3B` on the first 120, so their sets include train-split units. Seed 0 is excluded from every analysis by the loaders. `gpt-4o-mini`, `deepseek-v4-flash`, `qwen-plus` and `gpt-5-mini` carry three measurement seeds on their window arms (56, 99, 28 and 28 units); on `deepseek-v4-flash`'s 73 train-split units the surgery control and part of the seed-1 cor-

Table 5: Retraction and its placebos, blocking rung, paired on units: at j=1, retraction and the instruction-free placebo each against silent correction; at j=4, retraction against silent correction and against the instruction-free placebo (against the policy-restating placebo: Appendix A.3). Pooled: random effects; *: interval excludes zero. The instruction-free placebo was not run on `Qwen2.5-14B`, nor at j=4 on `qwen-plus`, and retraction not on `doubao-seed-2.0-lite`, `Granite-3.3-8B` or `Qwen3-30B-A3B`.

	j=1, against silent correction		j=4, retraction against	
policy	retraction	empty placebo	silent correction	empty placebo
`gpt-4o-mini`	−0.060 [−0.143, +0.024]	−0.071 [−0.155, +0.018]	**+0.117** [+0.015, +0.216]*	+0.107 [−0.013, +0.230]
`deepseek-v4-flash`	−0.064 [−0.192, +0.051]	**−0.192** [−0.333, −0.077]*	−0.058 [−0.250, +0.135]	**+0.218** [+0.090, +0.372]*
`gpt-4.1`	−0.095 [−0.333, +0.143]	−0.048 [−0.238, +0.143]	**+0.400** [+0.067, +0.667]*	**+0.385** [+0.154, +0.692]*
`qwen-plus`	−0.083 [−0.179, +0.000]	−0.071 [−0.167, +0.024]	−0.093 [−0.220, +0.020]	—
`gpt-5-mini`	−0.024 [−0.179, +0.131]	−0.048 [−0.167, +0.060]	+0.106 [−0.008, +0.235]	**+0.167** [+0.033, +0.317]*
`Qwen2.5-14B`	+0.058 [−0.077, +0.192]	—	−0.100 [−0.267, +0.067]	—
`Qwen3-14B`	+0.108 [−0.054, +0.270]	+0.027 [−0.135, +0.189]	+0.031 [−0.188, +0.250]	+0.161 [−0.032, +0.355]
pooled	−0.036 [−0.085, +0.013]	**−0.075** [−0.123, −0.027]*	+0.036 [−0.064, +0.136]	**+0.174** [+0.104, +0.245]*
I^2	0.10	0.00	0.64	0.00

rection at $j = 4$ were not run, so its surgery-adjusted contrasts rest on its 26 test-split units, and every other analysis on those of its 99 units that carry the cells it needs; `gpt-4.1` (21 units) and `Qwen2.5-14B` (52 units) carry one; `doubao-seed-2.0-lite` (23 units), `Qwen3-14B` (37), `Qwen3-30B-A3B` (58; its last 27 units only those cells) and `Granite-3.3-8B` (17) carry three on the control, fault, correction and surgery cells and one elsewhere (`Qwen3-14B`'s withheld record three). Seeds two and three were added after `gpt-4o-mini`'s single-seed window came out at +0.135, not significant (per-seed and first-seed estimates in Appendix A.6). The surgery control was first run at seed 1 only and later completed to every seed its correction cells carry; the withheld absent record carries the correction cells' seeds on `gpt-4o-mini`, `deepseek-v4-flash`, `gpt-5-mini` and `Qwen3-14B`, and seed 1 on `gpt-4.1` and `qwen-plus`. Task sets differ across policies, which is why cross-policy comparisons are generalisation claims and every within-policy contrast is paired on units. On the train-split units of the two API-served families also screened there, the window is **+0.257** [+0.143, +0.381]* (`gpt-4o-mini`) and **+0.323** [+0.215, +0.427]* (`deepseek-v4-flash`), so it does not rest on the 55 test tasks alone.

A.5 The customer simulator

The customer is `gpt-4o-mini` at temperature 0 in every API-served arm and in the `Qwen3-14B` (4- and 8-bit), `Qwen3-30B-A3B` and `Granite-3.3-8B` arms, the local `Qwen2.5-14B` in that model's own arms, and API-served `gpt-4o-mini` in its de-confound arms (§3.8), each of whose episodes is billed to that API, which confirms that the swap took effect.

The second customer. The control, fault, correction and surgery cells at $j \in \{1, 4\}$ were rerun at seeds 1–3 with a second customer on each policy's full task set: `deepseek-v4-flash` for `gpt-4o-mini` and `gpt-5-mini`, and `gpt-4.1-mini` for `gpt-4o-mini`. Table 6 compares them with the same cells and seeds under the default customer; a unit's window is averaged over the seeds at which both customers' two correction cells exist. The fault effect reproduces under every customer. Among the episodes that reach the late correction, the second customer's window is smaller for `gpt-4o-mini` under both second customers, and the three survivor differences pool to +0.182 [+0.095, +0.269] (I^2 = 0.00; Hartung–Knapp [−0.009, +0.373]). Keeping every scheduled episode at its observed outcome (intention to treat), the default customer's windows are **+0.196** [+0.107, +0.286]*, **+0.429** [+0.262, +0.595]* and **+0.196** [+0.107, +0.286]*, the second customer's **+0.131** [+0.030, +0.238]*, **+0.536** [+0.381, +0.690]* and **+0.161** [+0.060, +0.262]*, and the differences pool to +0.033 [−0.042, +0.109] (I^2 = 0.00; Hartung–Knapp [−0.133, +0.200]). Two of the three pairs share a policy, so these pools are not over independent families.

The withdrawal slope. One further quantity moves with the customer in the de-confound arm: on `Qwen2.5-14B`'s mild rung the within-unit slope of task success after a withdrawal is −0.054/turn with the API-served customer and +0.055/turn with the local one on the same units. Withdrawal leaves a hole and no replacement, so what the customer does next matters, and the withdrawal

Table 6: The correction window under a second customer, seeds 1–3, blocking rung: units reaching the $j = 4$ correction; the fault effect; the window and its surgery-adjusted form among episodes that reach the correction; and the default customer's window minus the second customer's, on the units with a window under both (survivors) and with every scheduled episode kept (ITT).

policy	customer	reached j=4	fault effect	window	surgery-adjusted	default − second	default − second, ITT
`gpt-4o-mini`	`gpt-4o-mini`	55/56	$\mathbf{-0.369}$ $[-0.470, -0.268]^*$	$\mathbf{+0.230}$ $[+0.127, +0.333]^*$	$\mathbf{+0.309}$ $[+0.176, +0.452]^*$	$\mathbf{+0.176}$ $[+0.064, +0.285]^*$	$+0.065$ $[-0.042, +0.173]$
	`deepseek-v4-flash`	54/56	$\mathbf{-0.357}$ $[-0.458, -0.256]^*$	$+0.068$ $[-0.034, +0.167]$	$+0.050$ $[-0.083, +0.190]$		
`gpt-5-mini`	`gpt-4o-mini`	26/28	$\mathbf{-0.798}$ $[-0.893, -0.690]^*$	$\mathbf{+0.391}$ $[+0.212, +0.571]^*$	$\mathbf{+0.380}$ $[+0.160, +0.587]^*$	$+0.037$ $[-0.287, +0.389]$	-0.107 $[-0.321, +0.107]$
	`deepseek-v4-flash`	19/28	$\mathbf{-0.881}$ $[-0.964, -0.774]^*$	$\mathbf{+0.377}$ $[+0.123, +0.623]^*$	$\mathbf{+0.583}$ $[+0.345, +0.798]^*$		
`gpt-4o-mini`	`gpt-4o-mini`	55/56	$\mathbf{-0.369}$ $[-0.470, -0.268]^*$	$\mathbf{+0.230}$ $[+0.127, +0.333]^*$	$\mathbf{+0.309}$ $[+0.176, +0.452]^*$	$\mathbf{+0.223}$ $[+0.070, +0.380]^*$	$+0.036$ $[-0.083, +0.161]$
	`gpt-4.1-mini`	51/56	$\mathbf{-0.286}$ $[-0.405, -0.173]^*$	-0.003 $[-0.128, +0.121]$	-0.053 $[-0.242, +0.125]$		
pooled (random)						$\mathbf{+0.182}$ $[+0.095, +0.269]^*$	$+0.033$ $[-0.042, +0.109]$

residuals of Table 9 (nine families with the `gpt-4o-mini` customer, `Qwen2.5-14B` with its own) should not be read as customer-independent.

Script degeneracy on the local arm. Small locally served policies sometimes drop out of the prompt language, which would be a mundane alternative explanation for `Qwen2.5-14B` having no window. We measured it. CJK script appears in the transcript (in the policy's own turns or the local simulated customer's) in 0.035 of that arm's $1{,}940$ blocking-rung episodes, against 0.0016 on `qwen-plus` and 0.0000 on the five other API arms. Its clean accuracy is 0.538 over all episodes and 0.520 over the ones free of such text, and dropping every task unit with an affected episode in the two correction cells moves the correction window from $+0.026$ $[-0.158, +0.211]$ ($n = 38$) to $+0.027$ $[-0.162, +0.216]$ ($n = 37$). Most affected episodes are in other cells of the design; in the two correction cells only one of the 38 window units has one, so the check shows that the window is not driven by degenerate episodes, not that the arm is free of them.

The 8-bit re-serving of `Qwen3-14B`. To test whether 4-bit weights remove the window, `Qwen3-14B` was re-served from its official FP8 checkpoint with everything else unchanged (chat template, thinking off, API-served customer) and run on the AWQ arm's 37 window units at seeds 1–3: the no-fault control, the fault, the corrections at $j \in \{1, 4\}$ and their surgery controls (666 episodes). The fault effect is $\mathbf{-0.207}$ $[-0.378, -0.036]^*$ (4-bit: $\mathbf{-0.261}$ $[-0.396, -0.135]^*$); the window is -0.083 $[-0.222, +0.056]$ (4-bit: -0.014 $[-0.126, +0.099]$), surgery-adjusted $\mathbf{-0.283}$ $[-0.515, -0.061]^*$, and -0.054 $[-0.189, +0.081]$ counting every scheduled episode; 4-bit minus 8-bit on the 36 units with both is $+0.079$ $[-0.102, +0.269]$. With 8-bit weights the late correction recovers no less than the early one.

A.6 Retention, per-seed contrasts, and the commitment channel

Table 7 gives, per arm on the blocking rung, the number of task units whose episode reached the scheduled intervention against the number in the design, and the share of corrected episodes in which the agent had already attempted a state-changing call, and made one the environment accepted, before the correction (at most 0.02 at $j = 1$). Survival is decided per episode: a unit enters a lagged cell with the seeds whose episode reached the intervention, and a paired contrast uses the seeds that survived in both of its cells. That is why the seed-averaged window can differ from the mean of the per-seed windows below (for `gpt-5-mini` it exceeds all three), and why a unit count of $26/26$ can hide a dropped episode. Keeping every episode at its observed outcome instead, with no survivor filter (an intention-to-treat window), gives $\mathbf{+0.196}$ $[+0.107, +0.292]^*$ on `gpt-4o-mini`, $\mathbf{+0.389}$ $[+0.300, +0.480]^*$ on `deepseek-v4-flash`, $\mathbf{+0.524}$ $[+0.286, +0.762]^*$ on `gpt-4.1`, $\mathbf{+0.190}$ $[+0.036, +0.357]^*$ on `qwen-plus`, $\mathbf{+0.429}$ $[+0.262, +0.595]^*$ on `gpt-5-mini`, $\mathbf{+0.319}$ $[+0.145, +0.493]^*$ on `doubao-seed-2.0-lite`, $+0.058$ $[-0.096, +0.192]$ on `Qwen2.5-14B`, $+0.009$ $[-0.099, +0.117]$ on `Qwen3-14B`, $+0.109$ $[+0.000, +0.218]$ on `Qwen3-30B-A3B` and -0.176 $[-0.373, +0.020]$ on `Granite-3.3-8B`; pooled $\mathbf{+0.326}$ $[+0.224, +0.428]^*$ over the API-served families ($I^2 = 0.69$) and $\mathbf{+0.200}$ $[+0.086, +0.314]^*$ over all ten ($I^2 = 0.86$). For a unit whose episode ended before turn four this compares the $j = 1$ correction with the uncorrected outcome, so it answers what a harness that schedules a correction at turn four would see, not how the survivors' window is biased.

Table 7: Blocking rung: units reaching the intervention over units designed; and, in the correction arm, the share of intervened episodes with a state-changing call attempted before the correction at $j = 2$ and $j = 4$, and with one the environment accepted at $j = 4$.

	correction, reached/designed			withdrawal, reached/designed			write attempted		accepted
policy	j=1	j=2	j=4	j=1	j=2	j=4	j=2	j=4	j=4
gpt-4o-mini	56/56	56/56	55/56	56/56	56/56	55/56	0.19	0.71	0.64
deepseek-v4-flash	99/99	99/99	92/99	99/99	99/99	87/99	0.00	0.25	0.25
gpt-4.1	21/21	21/21	17/21	21/21	21/21	18/21	0.05	0.18	0.18
qwen-plus	28/28	28/28	27/28	28/28	28/28	26/28	0.12	0.51	0.40
gpt-5-mini	28/28	28/28	26/28	28/28	28/28	26/28	0.08	0.34	0.34
doubao-seed-2.0-lite	23/23	21/23	19/23	23/23	23/23	14/23	0.00	0.58	0.56
Qwen2.5-14B	52/52	52/52	38/52	51/52	52/52	37/52	0.50	0.76	0.55
Qwen3-14B	37/37	36/37	37/37	37/37	36/37	32/37	0.33	0.67	0.48
Qwen3-30B-A3B	58/58	30/31	57/58	31/31	31/31	29/31	0.33	0.66	0.54
Granite-3.3-8B	17/17	17/17	17/17	17/17	17/17	15/17	0.29	0.63	0.57

Per-seed windows. On the three-seed arms the correction window (j=1 minus j=4) by seed is: gpt-4o-mini $+0.135$ $[-0.019, +0.288]$, $+0.189$ $[+0.038, +0.340]$, $+0.269$ $[+0.135, +0.404]$; deepseek-v4-flash $+0.327$ $[+0.163, +0.470]$, $+0.261$ $[+0.136, +0.386]$, $+0.337$ $[+0.217, +0.458]$; qwen-plus $+0.250$ $[+0.000, +0.500]$, $+0.160$ $[-0.040, +0.360]$, $+0.231$ $[+0.038, +0.423]$; gpt-5-mini $+0.381$ $[+0.143, +0.619]$, $+0.333$ $[+0.000, +0.611]$, $+0.318$ $[+0.091, +0.545]$; on its correction cells, doubao-seed-2.0-lite $+0.250$ $[+0.000, +0.500]$, $+0.200$ $[-0.067, +0.467]$, $+0.000$ $[-0.308, +0.308]$; and, locally served, Qwen3-14B -0.088 $[-0.294, +0.118]$, $+0.061$ $[-0.121, +0.242]$, -0.057 $[-0.229, +0.114]$, Qwen3-30B-A3B $+0.038$ $[-0.132, +0.208]$, $+0.151$ $[-0.019, +0.302]$, $+0.111$ $[-0.056, +0.278]$ and Granite-3.3-8B -0.188 $[-0.500, +0.125]$, -0.125 $[-0.438, +0.188]$, -0.071 $[-0.286, +0.143]$. On the five API-served arms fourteen of fifteen are positive and one is zero, eight with a lower bound above zero; on the three local arms five of nine are negative and none excludes zero. gpt-4.1 ($+0.471$ $[+0.235, +0.706]$) and Qwen2.5-14B ($+0.026$ $[-0.158, +0.211]$) carry one seed and were not escalated. Pooled over the first seed of every arm alone, before any seed was added, the window is $+\mathbf{0.163}$ $[+0.044, +0.282]^*$ over all ten ($I^2 = 0.70$; Hartung–Knapp $[+0.018, +0.308]$).

The commitment channel. The window recomputed on units with no state-changing call before either correction, against the unrestricted window on the same convention: gpt-4o-mini $+\mathbf{0.385}$ $[+0.179, +0.590]^*$ ($n = 26$) vs. $+\mathbf{0.230}$ $[+0.133, +0.339]^*$ ($n = 55$); deepseek-v4-flash $+\mathbf{0.335}$ $[+0.241, +0.429]^*$ (78) vs. $+\mathbf{0.319}$ $[+0.230, +0.409]^*$ (92); gpt-4.1 $+\mathbf{0.429}$ $[+0.214, +0.714]^*$ (14) vs. $+\mathbf{0.471}$ $[+0.235, +0.706]^*$ (17); qwen-plus $+0.020$ $[-0.157, +0.196]$ (17) vs. $+\mathbf{0.191}$ $[+0.043, +0.346]^*$ (27); gpt-5-mini $+\mathbf{0.484}$ $[+0.278, +0.683]^*$ (21) vs. $+\mathbf{0.391}$ $[+0.205, +0.564]^*$ (26); doubao-seed-2.0-lite $+0.117$ $[-0.117, +0.400]$ (10) vs. $+\mathbf{0.219}$ $[+0.018, +0.430]^*$ (19); Qwen2.5-14B $+0.000$ $[-0.333, +0.333]$ (9) vs. $+0.026$ $[-0.158, +0.211]$ (38); Qwen3-14B $-\mathbf{0.280}$ $[-0.515, -0.030]^*$ (22) vs. -0.014 $[-0.122, +0.099]$ (37); Qwen3-30B-A3B $+0.124$ $[-0.043, +0.290]$ (31) vs. $+\mathbf{0.126}$ $[+0.015, +0.237]^*$ (57); Granite-3.3-8B -0.185 $[-0.556, +0.222]$ (9) vs. -0.167 $[-0.363, +0.020]$ (17). Both columns use the bootstrap of Table 2 (same resampling seed), so the unrestricted column reproduces that table and the two differ only through the units included. Pooled (random effects): $+\mathbf{0.295}$ $[+0.156, +0.434]^*$ over the API-served families ($I^2 = 0.71$) and $+\mathbf{0.163}$ $[+0.017, +0.309]^*$ over all ten ($Q = 46.48$ on 9 df, $I^2 = 0.81$). Whether a write happens before turn four is itself behaviour under the fault, so this split describes; it does not identify.

Leave-one-family-out. Dropping each API-served family in turn and re-pooling the six, the random-effects window ranges from $+0.273$ $[+0.199, +0.346]$ (without gpt-5-mini) to $+0.307$ $[+0.223, +0.391]$ (without gpt-4o-mini), and the surgery-adjusted window from $+0.278$ $[+0.187, +0.369]$ (without gpt-5-mini) to $+0.324$ $[+0.233, +0.416]$ (without qwen-plus). Over all ten the raw window stays above zero whichever family is dropped (lowest $+0.151$ $[+0.046, +0.257]$, without gpt-4.1), but the surgery-adjusted one does not

(+0.121 [−0.016, +0.258] without `gpt-5-mini`). Leaving out a family cannot reveal a bias the families share, such as the customer nine of them have (Appendix A.5).

Share of the damage recovered. The share a correction buys back is $(c_j - f)/(b - f)$, with c_j success corrected at lag j, f under the never-corrected fault and b under the no-fault control, on the units all four cells share, so both lags are on one unit set; intervals are unit bootstraps of the ratio with the same draws at both lags, dropping the resamples whose denominator falls below 0.05, and a family whose observed denominator does, `Granite-3.3-8B` (0.049), is left out (the others' are 0.24 to 0.82). By family, at $j = 1$ and $j = 4$: `gpt-4o-mini` 0.58/−0.02, `deepseek-v4-flash` 0.73/0.33, `gpt-4.1` 0.71/0.14, `qwen-plus` 0.76/0.40, `gpt-5-mini` 0.57/0.08, `doubao-seed-2.0-lite` 0.82/0.47, `Qwen2.5-14B` 0.47/0.41, `Qwen3-14B` 0.45/0.50, `Qwen3-30B-A3B` 0.82/0.28. Pooled over the nine: $+\mathbf{0.678}$ $[+0.602, +0.754]^*$ and $+\mathbf{0.254}$ $[+0.132, +0.376]^*$, and the per-family declines pool to $+\mathbf{0.420}$ $[+0.317, +0.524]^*$ ($I^2 = 0.19$), which need not equal the difference of the two pooled shares; over the six API-served families $+\mathbf{0.690}$ $[+0.609, +0.771]^*$ and $+\mathbf{0.218}$ $[+0.076, +0.360]^*$, a decline of $+\mathbf{0.449}$ $[+0.361, +0.536]^*$. The share is not surgery-adjusted. In the working model of §2 the ratio of the shares is $(1 - \sigma)^3$: the pooled shares correspond to $\sigma \approx 0.28$, and per family σ is 0.24, 0.42, 0.19, 0.48, 0.17, 0.04, −0.03 and 0.30 (in the order above, `gpt-4o-mini` omitted), above the 0.37 at which the model's j=1 minus j=4 difference stops rising in two families and at zero in the two 14B `Qwen` models; `gpt-4o-mini`'s negative $j = 4$ share lies outside the model.

Primary and secondary analyses. The primary contrast, fixed with the design, is the survivor window (success corrected at $j = 1$ minus $j = 4$, a lagged cell keeping the episodes that reached the intervention) per family and pooled by DerSimonian–Laird over every family run; its within-unit trend test is the second primary test, and §3.1 gives both counts after a Holm correction. The surgery-adjusted, intention-to-treat, no-write and first-seed windows, the Hartung–Knapp intervals and the leave-one-out pools are sensitivity analyses of that contrast. The split into API-served and locally served families was not planned; it is reported because the local families turned out to be the heterogeneity, and pools restricted to either group are descriptive.

Small-k intervals. Where one contrast appears in two tables its interval can differ in the third decimal, because each analysis draws its own bootstrap resamples and some keep only the units a further cell shares. Every pool has two to ten families, so the DerSimonian–Laird intervals undercover and I^2 is imprecise. Table 8 gives, for the pools the text relies on, the modified Hartung–Knapp interval (the random-effects standard error rescaled by the observed dispersion, floored at one as in Röver et al., 2015, with a t quantile on $k - 1$ df) and the test-based 95% interval for I^2 (Higgins & Thompson, 2002). Over the API-served families the window and its surgery-adjusted and intention-to-treat forms, the withheld window, the falsity excess (raw, surgery-adjusted and ITT) and corrected minus withheld at $j = 4$ keep their sign; over all ten the raw and ITT windows keep it, but the surgery-adjusted and no-write windows and the falsity excess with `Qwen3-14B` do not. The recovered shares and their decline, the rung gap over six families, the free-absent contrast at $j = 4$, the empty placebo's cost at $j = 1$ and the retraction's gain over it at $j = 4$ keep their sign; the append-only window, the retraction's $j = 1$ cost, the rung gap without `gpt-4o-mini`, both re-read slope gaps, the customer contrasts and the rewind's clean cost and net shortfall do not. A 95% prediction interval for a new family (Higgins et al., 2009) is $[-0.188, +0.540]$ for the raw window over all ten and $[+0.124, +0.450]$ over the API-served families. The order-record inert effect is equivalent to zero within ± 0.20 under both intervals; for retraction minus the policy-restating placebo at $j = 4$ the 90% intervals are $[-0.055, +0.054]$ (DL) and $[-0.076, +0.075]$ (Hartung–Knapp), inside ± 0.10.

A.7 Withdrawal

Table 9 gives the withdrawal residuals and re-read probabilities behind §3.3.

A.8 Re-reading on both rungs

Figure 3 and Table 10 give the re-read curves and slopes behind §3.5, for the six families with a mild-rung withdrawal design. Fault success here is on the units of the re-read analysis, so it can differ from Table 3.

Table 8: Pools under small-k corrections: DerSimonian–Laird estimate and interval as in the main text, the modified Hartung–Knapp interval, and I^2 with its 95% interval.

pool	k	DL	Hartung–Knapp	I^2 [interval]
correction window, API-served	6	$+0.287$ $[+0.217, +0.358]$	$[+0.193, +0.382]$	0.29 $[0.00, 0.71]$
surgery-adjusted window, API-served	6	$+0.294$ $[+0.210, +0.377]$	$[+0.184, +0.403]$	0.00 $[0.00, 0.63]$
intention-to-treat window, API-served	6	$+0.326$ $[+0.224, +0.428]$	$[+0.193, +0.460]$	0.69 $[0.26, 0.87]$
correction window, all ten	10	$+0.176$ $[+0.071, +0.281]$	$[+0.043, +0.309]$	0.81 $[0.67, 0.90]$
surgery-adjusted window, all ten	10	$+0.148$ $[+0.015, +0.281]$	$[-0.005, +0.302]$	0.76 $[0.55, 0.87]$
intention-to-treat window, all ten	10	$+0.200$ $[+0.086, +0.314]$	$[+0.053, +0.347]$	0.86 $[0.77, 0.92]$
share recovered, j=1	9	$+0.678$ $[+0.602, +0.754]$	$[+0.589, +0.767]$	0.00 $[0.00, 0.58]$
share recovered, j=4	9	$+0.254$ $[+0.132, +0.376]$	$[+0.111, +0.397]$	0.63 $[0.24, 0.82]$
decline in share, j=1 − j=4	9	$+0.420$ $[+0.317, +0.524]$	$[+0.293, +0.548]$	0.19 $[0.00, 0.61]$
withheld window, API-served	5	$+0.129$ $[+0.071, +0.187]$	$[+0.047, +0.211]$	0.00 $[0.00, 0.76]$
falsity excess, API-served (Table 11)	5	$+0.180$ $[+0.082, +0.279]$	$[+0.041, +0.320]$	0.00 $[0.00, 0.78]$
falsity excess, surgery-adjusted, API-served	5	$+0.202$ $[+0.099, +0.305]$	$[+0.056, +0.348]$	0.00 $[0.00, 0.66]$
falsity excess, ITT, API-served	5	$+0.192$ $[+0.077, +0.307]$	$[+0.026, +0.358]$	0.25 $[0.00, 0.70]$
falsity excess, with `Qwen3-14B` (6)	6	$+0.143$ $[+0.016, +0.270]$	$[-0.027, +0.314]$	0.45 $[0.00, 0.78]$
corrected − withheld, j=4, API-served	5	-0.281 $[-0.459, -0.103]$	$[-0.540, -0.022]$	0.76 $[0.42, 0.90]$
mild-rung window	6	$+0.072$ $[-0.007, +0.152]$	$[-0.032, +0.176]$	0.00 $[0.00, 0.73]$
rung gap, within units	6	$+0.206$ $[+0.097, +0.314]$	$[+0.063, +0.348]$	0.00 $[0.00, 0.68]$
rung gap, without `gpt-4o-mini`	5	$+0.183$ $[+0.050, +0.316]$	$[-0.005, +0.371]$	0.00
re-read slope gap, six families	6	-0.071 $[-0.134, -0.009]$	$[-0.153, +0.011]$	0.78 $[0.53, 0.90]$
re-read slope gap, five with milder mild rung	5	-0.090 $[-0.152, -0.028]$	$[-0.180, +0.000]$	0.71 $[0.27, 0.89]$
inert fault effect, order record	10	-0.050 $[-0.114, +0.013]$	$[-0.124, +0.023]$	0.00 $[0.00, 0.16]$
append-only window	5	$+0.146$ $[+0.038, +0.255]$	$[-0.008, +0.300]$	0.53 $[0.00, 0.83]$
corrected − free absent, j=4	5	-0.323 $[-0.464, -0.183]$	$[-0.532, -0.114]$	0.65 $[0.07, 0.87]$
retraction − correction, j=1	7	-0.036 $[-0.085, +0.013]$	$[-0.098, +0.025]$	0.10 $[0.00, 0.74]$
retraction − correction, j=1, placebo families	6	-0.051 $[-0.099, -0.002]$	$[-0.114, +0.013]$	0.00 $[0.00, 0.72]$
retraction − correction, j=2	6	-0.017 $[-0.083, +0.050]$	$[-0.105, +0.071]$	0.35 $[0.00, 0.74]$
empty placebo − correction, j=1	6	-0.075 $[-0.123, -0.027]$	$[-0.138, -0.012]$	0.00 $[0.00, 0.73]$
retraction − correction, j=4	7	$+0.036$ $[-0.064, +0.136]$	$[-0.100, +0.172]$	0.64 $[0.19, 0.84]$
retraction − placebo, j=4	7	-0.001 $[-0.065, +0.064]$	$[-0.096, +0.095]$	0.38 $[0.00, 0.74]$
empty placebo − correction, j=4	5	-0.061 $[-0.135, +0.013]$	$[-0.175, +0.052]$	0.39 $[0.00, 0.77]$
retraction − empty placebo, j=4	5	$+0.174$ $[+0.104, +0.245]$	$[+0.075, +0.274]$	0.00 $[0.00, 0.79]$
rewind: clean cost	3	-0.155 $[-0.238, -0.071]$	$[-0.337, +0.028]$	0.00
rewind: gain net of clean cost	3	$+0.161$ $[-0.018, +0.340]$	$[-0.253, +0.574]$	0.55 $[0.00, 0.87]$
rewind: shortfall against j=1, net	3	$+0.118$ $[-0.036, +0.272]$	$[-0.220, +0.456]$	0.41 $[0.00, 0.82]$
customer: default − second, survivors	3	$+0.182$ $[+0.095, +0.269]$	$[-0.009, +0.373]$	0.00 $[0.00, 0.79]$
customer: default − second, ITT	3	$+0.033$ $[-0.042, +0.109]$	$[-0.133, +0.200]$	0.00 $[0.00, 0.89]$

Table 9: Withdrawal on the blocking rung: success minus the arrangement-matched no-fault control; $P(\mathrm{rr})$, the probability the agent re-reads the deleted record; and the least-squares slope of $P(\mathrm{rr})$ per turn over $j \in \{0, 1, 2, 4\}$, with its R^2 (Table 10 fits $j \in \{1, 2, 4\}$, so its slopes can differ in either direction). The residuals include the deletion's own cost, which its surgery control puts at up to 0.250 at $j = 1$ (§3.1). The $j = 4$ cells are survivor subsamples (Table 7).

	residual vs. no-fault control				$P(\mathrm{rr})$				slope	R^2
policy	j=0	j=1	j=2	j=4	j=0	j=1	j=2	j=4		
`gpt-4o-mini`	-0.262^*	-0.387^*	-0.440^*	-0.339^*	0.70	0.45	0.30	0.27	-0.100	0.77
`deepseek-v4-flash`	-0.226^*	-0.566^*	-0.710^*	-0.751^*	0.75	0.45	0.24	0.14	-0.146	0.86
`gpt-4.1`	-0.048	-0.381^*	-0.524^*	-0.833^*	0.95	0.62	0.38	0.06	-0.218	0.97
`qwen-plus`	-0.155^*	-0.429^*	-0.405^*	-0.564^*	0.73	0.45	0.43	0.32	-0.089	0.78
`gpt-5-mini`	-0.155	-0.357^*	-0.482^*	-0.744^*	0.83	0.67	0.49	0.21	-0.157	1.00
`doubao-seed-2.0-lite`	-0.174	-0.435^*	-0.522^*	-0.500^*	0.83	0.39	0.13	0.21	-0.140	0.59
`Qwen2.5-14B`	-0.058	-0.392^*	-0.365^*	-0.459^*	0.48	0.18	0.23	0.08	-0.084	0.70
`Qwen3-14B`	-0.162	-0.243^*	-0.333^*	-0.375^*	0.81	0.57	0.44	0.41	-0.094	0.77
`Qwen3-30B-A3B`	-0.129	-0.161	-0.194	-0.172	0.68	0.52	0.32	0.28	-0.100	0.85
`Granite-3.3-8B`	-0.235	-0.059	-0.235^*	-0.067	0.71	0.47	0.12	0.13	-0.144	0.75

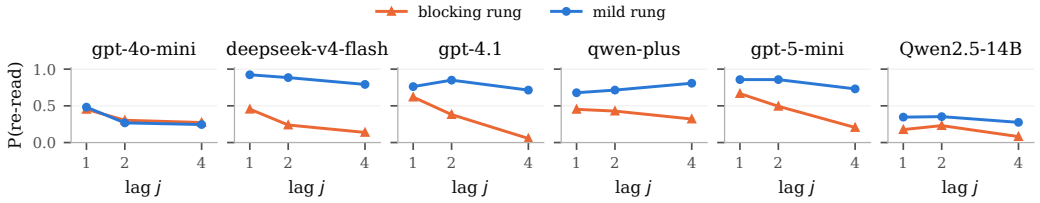


Figure 3: Probability that the agent re-reads the order record after its corrupted version is withdrawn at lag j, on the blocking rung (▲) and the mild rung (●) of the same task units.

Table 10: Re-reading on the two rungs: fault success, the slope of $P(\mathrm{rr})$ per turn in j on each rung (each on its own units), their paired within-unit difference (blocking minus mild, on the units both rungs share, so it can differ from the difference of the two slopes), and the within-unit window gap of §3.5 (blocking minus mild window). Pooled: random effects.

policy	succ., blocking	succ., mild	slope, blocking	slope, mild	paired difference	window gap
`deepseek-v4-flash`	0.037	0.423	−0.109	−0.042	**−0.111** [−0.173, −0.048]*	+0.269 [−0.026, +0.558]
`gpt-4.1`	0.000	0.190	−0.187	−0.007	**−0.179** [−0.266, −0.095]*	**+0.412** [+0.118, +0.765]*
`gpt-5-mini`	0.012	0.429	−0.156	−0.055	**−0.105** [−0.168, −0.045]*	+0.160 [−0.147, +0.474]
`qwen-plus`	0.190	0.536	−0.054	+0.025	−0.082 [−0.183, +0.019]	+0.045 [−0.212, +0.321]
`Qwen2.5-14B`	0.103	0.231	−0.052	−0.035	+0.023 [−0.051, +0.097]	+0.091 [−0.212, +0.394]
`gpt-4o-mini`	0.375	0.375	−0.053	−0.062	+0.012 [−0.044, +0.069]	**+0.252** [+0.069, +0.447]*
pooled, the five families whose mild rung was milder ($I^2 = 0.71$ and 0.00)					**−0.090** [−0.152, −0.028]*	**+0.183** [+0.050, +0.316]*
pooled, all six ($I^2 = 0.78$ and 0.00)					**−0.071** [−0.134, −0.009]*	**+0.206** [+0.097, +0.314]*

A.9 The withheld absent record

Table 11 gives the per-family contrasts behind §3.2. The bound there assumes that the withheld arm's error messages do not cost more by turn four than by turn one. A descriptive check (`code/withheld_errors.py`) finds no sign that they derail the agent: among withheld episodes at $j = 4$, those in which the agent had received at least one error succeed at 0.640, 0.485, 0.800 and 0.561 on `gpt-4o-mini`, `deepseek-v4-flash`, `qwen-plus` and `gpt-5-mini`, against 0.346, 0.533, 0.167 and 0.278 for those that had received none (18 of `gpt-4.1`'s 19 episodes had one). The split is post-treatment, since the agents that go looking for the record are also the ones that investigate, so it shows that errors do not go with lower success, not that they cost nothing.

Table 11: False premise versus withheld absent record (T11), on the units and seeds the four cells share (`deepseek-v4-flash`'s 26 test-split units, one seed on `qwen-plus`, one unit fewer on `gpt-4.1` and `Qwen3-14B`, so their corrected windows differ from Table 2; the four-cell bootstrap also gives other intervals): how often the agent tried to read the missing record before it arrived at $j = 4$; each arm's window (j=1 minus j=4); and corrected minus withheld window. Pooled: random effects.

policy	tried, j=4	withheld window	corrected window	difference
`gpt-4o-mini`	0.52	**+0.091** [+0.003, +0.185]*	**+0.230** [+0.127, +0.339]*	+0.139 [−0.018, +0.297]
`deepseek-v4-flash`	0.42	**+0.192** [+0.103, +0.288]*	**+0.308** [+0.147, +0.474]*	+0.115 [−0.058, +0.295]
`gpt-4.1`	0.95	+0.125 [−0.188, +0.438]	**+0.500** [+0.250, +0.750]*	+0.375 [+0.000, +0.750]
`qwen-plus`	0.56	+0.125 [−0.083, +0.333]	+0.250 [+0.000, +0.500]	+0.125 [−0.250, +0.458]
`gpt-5-mini`	0.79	+0.032 [−0.160, +0.231]	**+0.391** [+0.218, +0.571]*	**+0.359** [+0.109, +0.609]*
pooled, API-served (5)		**+0.129** [+0.071, +0.187]*	**+0.309** [+0.220, +0.398]*	**+0.180** [+0.082, +0.279]*
I^2		0.00	0.27	0.00
`Qwen3-14B`	0.53	+0.060 [−0.083, +0.199]	−0.023 [−0.134, +0.093]	−0.083 [−0.278, +0.125]
pooled, with `Qwen3-14B` (6)		**+0.119** [+0.065, +0.173]*	**+0.261** [+0.111, +0.412]*	**+0.143** [+0.016, +0.270]*
I^2		0.00	0.81	0.45

A.10 The absent record, free to fetch

In the first absent arm (T10) the target order is not in the prefix, and nothing stops the agent from calling `get_order_details` on it during its live turns. Table 12 gives the share of intervened

episodes in which the agent had fetched the record itself before the harness supplied it at turn j, success in the episodes where it had not, the within-unit slope of the absent curve, and corrected minus absent at each lag. The share rises with j in every family, and by $j = 4$ the agent has fetched the record itself in a majority of episodes in four of five families, so the flat curve is in large part the agent's own doing: a missing record is noticed and fetched. That is why the withheld variant (T11) carries the comparison in §3.2. On this arm the corrected false premise is not distinguishable from never having had the record at $j = 1$ (pooled -0.040 $[-0.137, +0.057]$) and is lower by $\mathbf{-0.323}$ $[-0.464, -0.183]^*$ at $j = 4$, significantly in five of five; the absent curve pools to j=1 minus j=4 of $+0.024$ $[-0.025, +0.072]$, with `gpt-5-mini` the one family whose absent curve also declines.

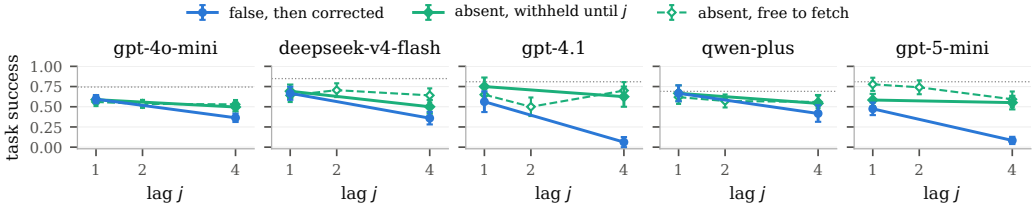


Figure 4: The true record arriving at turn j: after a false version was read and acted on (●, correction), or after the record was absent from the prefix, withheld until turn j (⧫, run at $j \in \{1, 4\}$; it and the correction on the units of Table 11) or free to be fetched (◊, dashed, on its own units, so gaps read off the plot are not the paired contrasts of Table 12). Dotted: no-fault control. Error bars: s.e. over units. The five API-served families that carry both absent arms; `Qwen3-14B`'s withheld arm is in Table 11.

Table 12: The absent record free to fetch (T10): share of episodes in which the agent fetched it before turn j; success in the remaining episodes at j=4 (count); the within-unit slope per turn of the absent curve (p), per family, with the pooled entry† given instead as j=1 minus j=4 (a positive value is a decline); and corrected minus absent at $j = 1$ and $j = 4$.

policy	fetched, j=1	j=2	j=4	success, no fetch	absent trend (p)	corr. − absent, j=1	j=4
`gpt-4o-mini`	0.06	0.38	0.53	0.29 (78)	−0.009 (0.29)	+0.018 [−0.083, +0.125]	**−0.182** [−0.339, −0.030]*
`deepseek-v4-flash`	0.32	0.36	0.42	0.56 (45)	−0.005 (0.38)	+0.026 [−0.141, +0.179]	**−0.282** [−0.474, −0.083]*
`gpt-4.1`	0.52	0.71	0.90	0.50 (2)	+0.029 (0.73)	+0.048 [−0.238, +0.333]	**−0.588** [−0.824, −0.353]*
`qwen-plus`	0.24	0.51	0.66	0.30 (27)	−0.018 (0.19)	−0.060 [−0.202, +0.071]	**−0.207** [−0.360, −0.053]*
`gpt-5-mini`	0.50	0.68	0.89	0.00 (3)	−0.063*(0.018)	**−0.286** [−0.500, −0.071]*	**−0.450** [−0.650, −0.250]*
pooled (random)					+0.024 [−0.025, +0.072]†	−0.040 [−0.137, +0.057]	**−0.323** [−0.464, −0.183]*

A.11 Append-only correction

A harness that learns a row was stale can re-issue the call and append the result without rewriting the transcript. An append-only arm appends the true record at turn j and leaves the false one in place (it has no surgery control of its own). Its window pools to $\mathbf{+0.146}$ $[+0.038, +0.255]^*$ over five families, but not under Hartung–Knapp ($[-0.008, +0.300]$). Leaving the false record in place changes success by $\mathbf{-0.171}$ $[-0.245, -0.098]^*$ against deletion at $j = 1$ ($I^2 = 0.23$), and on average little at $j = 4$ (-0.015 $[-0.127, +0.097]$, $I^2 = 0.76$, with `deepseek-v4-flash` the exception): early, deleting the false record as well is worth doing.

Table 13 gives, per family on the blocking rung, the append-only window (j=1 minus j=4) and append-only minus the standard correction (deletion plus append) at each lag. The arm ran on `gpt-4o-mini`, `deepseek-v4-flash`, `gpt-4.1`, `qwen-plus` and `gpt-5-mini`, at the correction cells' seeds, with no surgery control of its own; like every lagged cell, its $j = 4$ cell keeps only episodes that reached the intervention, so each column is on its own units.

A.12 Edits at the late correction

T12 and T13 are run at $j = 4$ on seeds 1 and 2 of three families (the combined rewind and reset on two, the full reset on `deepseek-v4-flash`), on the units that reached turn four in every cell of each comparison. *Rewind*: at turn 4 every message after the prefix is removed from the agent's

Table 13: Append-only correction: the true record is appended at turn j and the false observation is left in the transcript.

policy	window (j=1 − j=4)	append-only − correction, j=1	j=4
gpt-4o-mini	+0.003 [−0.105, +0.111]	**−0.173** [−0.268, −0.083]*	+0.009 [−0.082, +0.104]
deepseek-v4-flash	**+0.200** [+0.047, +0.360]*	**−0.333** [−0.500, −0.167]*	**−0.267** [−0.413, −0.120]*
gpt-4.1	+0.222 [+0.000, +0.444]	−0.143 [−0.429, +0.143]	+0.125 [+0.000, +0.312]
qwen-plus	+0.131 [−0.042, +0.298]	−0.083 [−0.250, +0.083]	−0.019 [−0.136, +0.086]
gpt-5-mini	**+0.269** [+0.077, +0.455]*	**−0.131** [−0.262, −0.012]*	+0.073 [−0.040, +0.200]
pooled (random)	**+0.146** [+0.038, +0.255]*	**−0.171** [−0.245, −0.098]*	−0.015 [−0.127, +0.097]
I^2	0.53	0.23	0.76

context, the false record is deleted and the true one appended, exactly as in the correction arm; the agent then sees its system prompt, the customer's opening message, the prefix and the true record, while the customer simulator keeps its own history. *Scrub*: the correction plus every live reply the agent wrote to the customer replaced by "(earlier reply removed by the harness)"; tool calls, tool results and the customer's replies stay. *Reset*: the correction plus the database reloaded to its initial state, which, the prefix being read-only, is its state at the end of the prefix; every write the agent made in its live turns is undone, and the transcript still records them. *Full reset*: the rewind and the reset, and the customer simulator's history cut back to its system prompt, the harness's greeting and its own opening message, so that nothing of the live turns survives except the steps they used from the budget. Each edit is also applied to a clean prefix at the same turn, and the surgery controls of the two plain corrections are in every comparison, so an edit's own cost can be taken out: its *net gain* is (edited minus plain late correction) minus (the same on a clean prefix), and its *net shortfall* is the early correction's surgery-adjusted value minus the edited late correction's. The *residual* is faulted minus clean success under the same edit; without an edit it is the late correction's residual damage.

Table 14: Edits at the $j = 4$ correction, seeds 1–2, blocking rung, on each comparison's shared units: success with the edit on a faulted and on a clean prefix; the residual (faulted minus clean) with the edit and, on the same units, after the plain late correction; the edit's gain over the plain late correction net of its clean-prefix cost; and its shortfall against the correction at $j = 1$, each net of its clean-prefix counterpart. Pooled: random effects.

policy	edit	n	faulted	clean	residual, edit	residual, plain j=4	net gain	net shortfall
gpt-4o-mini	scrub	51	0.431	0.686	**−0.255** [−0.382, −0.127]*	**−0.284** [−0.422, −0.147]*	+0.029 [−0.137, +0.206]	**+0.206** [+0.069, +0.343]*
gpt-4o-mini	rewind	51	0.451	0.539	−0.088 [−0.216, +0.039]	**−0.284** [−0.422, −0.147]*	**+0.196** [+0.049, +0.353]*	+0.039 [−0.108, +0.176]
gpt-4o-mini	reset	49	0.112	0.133	−0.020 [−0.133, +0.092]	**−0.286** [−0.429, −0.143]*	**+0.265** [+0.102, +0.439]*	−0.041 [−0.204, +0.122]
gpt-4o-mini	rewind + reset	51	0.392	0.392	+0.000 [−0.127, +0.127]	**−0.245** [−0.382, −0.108]*	**+0.245** [+0.059, +0.431]*	−0.059 [−0.216, +0.108]
deepseek-v4-flash	scrub	24	0.146	0.812	**−0.667** [−0.833, −0.479]*	**−0.417** [−0.604, −0.229]*	**−0.250** [−0.458, −0.042]*	**+0.479** [+0.271, +0.667]*
deepseek-v4-flash	rewind	24	0.208	0.667	**−0.458** [−0.646, −0.271]*	**−0.458** [−0.646, −0.271]*	+0.000 [−0.188, +0.188]	**+0.271** [+0.062, +0.479]*
deepseek-v4-flash	reset	23	0.217	0.087	+0.130 [−0.022, +0.304]	**−0.435** [−0.630, −0.239]*	**+0.565** [+0.304, +0.804]*	**−0.283** [−0.522, −0.043]*
deepseek-v4-flash	rewind + reset	26	0.385	0.788	**−0.404** [−0.558, −0.250]*	**−0.365** [−0.539, −0.192]*	−0.038 [−0.250, +0.154]	**+0.212** [+0.019, +0.423]*
deepseek-v4-flash	full reset	22	0.818	0.886	−0.068 [−0.136, +0.000]	**−0.455** [−0.659, −0.250]*	**+0.386** [+0.159, +0.614]*	−0.182 [−0.409, +0.045]
gpt-5-mini	scrub	18	0.167	0.750	**−0.583** [−0.806, −0.333]*	**−0.667** [−0.861, −0.444]*	+0.083 [−0.278, +0.417]	+0.167 [−0.083, +0.417]
gpt-5-mini	rewind	18	0.306	0.667	**−0.361** [−0.639, −0.083]*	**−0.722** [−0.889, −0.528]*	**+0.361** [+0.056, +0.667]*	+0.056 [−0.250, +0.361]
gpt-5-mini	reset	19	0.105	0.289	**−0.184** [−0.342, −0.053]*	**−0.658** [−0.842, −0.447]*	**+0.474** [+0.211, +0.711]*	−0.211 [−0.447, +0.026]
pooled ($k = 3$)	scrub						−0.060 [−0.268, +0.147]	**+0.285** [+0.097, +0.474]*
pooled ($k = 3$)	rewind						+0.161 [−0.018, +0.340]	+0.118 [−0.036, +0.272]
pooled ($k = 3$)	reset						**+0.415** [+0.227, +0.603]*	**−0.156** [−0.309, −0.003]*
pooled ($k = 2$)	rewind + reset						+0.107 [−0.171, +0.385]	+0.069 [−0.196, +0.334]

The reset alone is not interpretable: undoing a clean episode's legitimate writes while its transcript still reports them costs **−0.599** [−0.742, −0.455]* pooled, and faulted episodes collapse with them, so its large net gain is an artefact of subtracting that cost. The rewind's clean cost is **−0.155** [−0.238, −0.071]*. With the reset added to the rewind the clean cost falls to −0.019 on deepseek-v4-flash and rises to −0.294 on gpt-4o-mini; on the latter, faulted and clean episodes then succeed equally often.

B NEGATIVE RESULTS AND WITHDRAWN CLAIMS

Each of the following was a claim we were prepared to make, or an instrument we intended to use, until one of our own controls or tests removed it; the last two paragraphs are scope notes. We report them so that the tests the surviving claims passed are visible.

"Correction at $j = 1$ fully restores the agent." Withdrawn. The first version of the window analysis found the corrected-at-j=1 cell statistically indistinguishable from the no-fault control and reported that as restoration, which reads a failure to reject as equivalence. Two one-sided tests at a margin of $\delta = 0.20$, fixed after that analysis and before the tests were run, fail to establish equivalence in nine of the ten arms (90% intervals throughout): the residual is -0.173 $[-0.262, -0.083]$ on `gpt-4o-mini`, -0.189 $[-0.259, -0.118]$ on `deepseek-v4-flash`, -0.143 $[-0.333, +0.048]$ on `gpt-4.1`, -0.131 $[-0.286, +0.012]$ on `qwen-plus`, -0.298 $[-0.429, -0.179]$ on `gpt-5-mini`, -0.072 $[-0.217, +0.072]$ on `doubao-seed-2.0-lite`, -0.288 $[-0.442, -0.135]$ on `Qwen2.5-14B`, -0.144 $[-0.234, -0.054]$ on `Qwen3-14B` and -0.196 $[-0.353, -0.039]$ on `Granite-3.3-8B`; the exception is `Qwen3-30B-A3B`, -0.046 $[-0.115, +0.017]$, whose correction at $j = 1$ is equivalent to the no-fault control. Correction at one turn is much better than at four in the API-served families, and it is not shown to be complete in any of them; in six families the 90% interval lies below zero.

"An explicit retraction is a remedy." Withdrawn. With the first four arms (`gpt-4o-mini`, `deepseek-v4-flash`, `gpt-4.1`, `Qwen2.5-14B`) the retraction-minus-correction contrast at $j = 4$ pooled positive under fixed effects and was carried by `gpt-4.1`; with seven arms it is $+0.036$ $[-0.064, +0.136]$ (random effects; $I^2 = 0.64$; per family in Table 5), two families significantly positive and three pointing the other way, and against the policy-restating placebo it is -0.001 $[-0.065, +0.064]$ (Appendix A.3). On `gpt-4.1` alone the retraction is worth $+\mathbf{0.400}$ $[+0.067, +0.667]^*$ against silent correction and $+\mathbf{0.357}$ $[+0.143, +0.643]^*$ against the placebo, which points to its content on that policy; on `gpt-4o-mini` its $+\mathbf{0.117}$ $[+0.015, +0.216]^*$ against silent correction is matched by the placebo. A benefit on one or two of seven policies does not support a general remedy. A second reading, that the retraction's cost at $j = 1$ was the cost of being told, was withdrawn when the placebo was run at $j = 1$: the placebo costs as much (§3.4).

"Susceptibility rises with capability." Withdrawn; it fails its own floor check. On the blocking rung three of the four policies in the original capability comparison (seed 1) sit below 0.20 under the fault (`Qwen2.5-14B` 0.096, `deepseek-v4-flash` 0.020, `gpt-4.1` 0.000, on that seed's units), so their absolute drop is capped by their clean score and any correlation of the drop with clean accuracy is a floor artefact; regressing the *relative* drop (drop over clean accuracy) on clean accuracy over the same four policies gives slope $+0.455$ and $r = +0.30$, which supports neither direction. The arms added later do not settle it either way: `qwen-plus` at 0.690 clean loses 0.500, more than `gpt-4o-mini` at 0.744 (0.369), while `gpt-5-mini` at 0.810 loses 0.798 (Tables 1 and 3). The locally served families add little: `Qwen3-14B` at 0.577 clean loses 0.261, `Qwen3-30B-A3B` at 0.701 loses 0.230, `Granite-3.3-8B` at 0.608 loses 0.059. Chen et al. (2026b) report the correlation but attribute it to the same floor, and Wang et al. (2026b) report none; the question stays open at ten policies.

The premise in the agent's own words. A deterministic detector scans the agent's own `respond` turns for the injected value, calibrated on the no-fault control to a false-positive rate of 0.02–0.14 per policy; it was run on `gpt-4o-mini`, `deepseek-v4-flash`, `gpt-4.1`, `qwen-plus` and `gpt-5-mini`. The premise does enter the agent's words in 0.67–0.97 of never-corrected faulted episodes, so it is not that the agent silently acts on a value it never states. But the rate of *still* asserting it after a correction does not order with j consistently across policies, and the differences are small and of both signs (post-correction assertion at j=1 versus j=4: `gpt-4o-mini` 0.41 vs 0.27; `deepseek-v4-flash` 0.54 vs 0.51; `gpt-4.1` 0.10 vs 0.12; `qwen-plus` 0.30 vs 0.47; `gpt-5-mini` 0.13 vs 0.07), and `gpt-4.1` fails at $j = 4$ without verbally restating the premise. The failure that persists is therefore more than a repetition of the false value.

An LLM judge of the latent state. A judge model was asked, on 2,806 trajectories of the `gpt-4o-mini` and `Qwen2.5-14B` arms, whether the agent held a false premise. Against the ground truth the design supplies (a faulted trajectory contains one, a clean trajectory does not) its sensitivity was 0.69–0.74 at a false-positive rate of 0.58–0.60, and a cell-level fit relating the judged false-premise rate to task success reached R^2 of 0.13 and 0.013 on the two arms. It was abandoned, which is why the causal claims in the paper are identified by intervention (performing the re-read

for the agent, deleting the record, swapping the rung, the customer or the transcript), and the re-read rates are reported as descriptive.

The age–budget dissociation. An early hypothesis held that the contamination age k (how many prefix reads separate the corrupted observation from the first live turn) and the live-step budget would dissociate, with a tighter budget (the clean step count plus three, against the default 20) hurting old contamination more. Net of what the budget does to a clean episode, the generous budget's effect at $k = 7$ is $+0.18$ $[+0.00, +0.36]$ on `Qwen2.5-14B`'s mild rung (50 units) and $\mathbf{-0.333}$ $[-0.619, -0.048]^*$ on `gpt-4o-mini`'s blocking rung, and at $k = 0$ neither differs from zero. The two arms disagree in sign, so no dissociation is claimed.

A contamination-position law. The fault effect is significant at every arrangement in all five position sweeps ($k \in \{0, 1, 2, 4, 7\}$), but its dependence on k is not consistent: over the three sweeps with headroom the slopes in k are -0.027 (`gpt-4o-mini`, blocking; interval excluding zero), -0.014 (`Qwen2.5-14B`, mild) and $+0.029$ (`deepseek-v4-flash`, mild), pooling to -0.010 $[-0.026, +0.006]$ with $Q = 7.09$ on 2 df. Position within the prefix is not a law we can state, and `gpt-4o-mini`'s negative slope means the window measured at $k = 0$, the most recent read, need not transfer to older reads; every window arm holds it fixed at $k = 0$, so the design does not separate the lag since the agent started acting on the premise from the time since it was read; against the positional sensitivity of Liu et al. (2024), note that the prefix here is at most nine short tool results, well inside any policy's effective context.

A surface-versus-context axis taxonomy. An attempt to classify faults by whether they change the surface form of the observation (terse tool documentation, sanitised errors, a text-based tool protocol) or its relation to context (truncating stale observations) was carried by a single axis in one arm and flipped sign in another, so no taxonomy is claimed. The one axis a harness designer might act on, truncating observations older than a few turns, changes success under the false premise by -0.12 $[-0.26, +0.02]$ on `Qwen2.5-14B`'s mild rung, $+0.06$ $[-0.06, +0.20]$ on its blocking rung and -0.036 $[-0.197, +0.125]$ on `gpt-4o-mini`: no effect in either direction is established.

The second environment. τ-bench airline was ported and its ground-truth replays validated offline. Of the 14 tasks whose ground truth references a reservation that can carry the fault, `gpt-4o-mini` solves 0 at the clean screen, `deepseek-v4-flash` 1, `gpt-4.1` 2 and `gpt-5-mini` 3; `gpt-4.1` also solves 2 of 14 with no prefix at all, so the prefix is not the cause, though the port as a whole is not independently checked against published airline scores. On the clean airline screen `gpt-5-mini` transfers to a human agent in 7 of its 14 episodes, `deepseek-v4-flash` in 2 and `gpt-4.1` in 1. A within-environment contrast is not supportable at this scale and none is claimed.

Families dropped at or before the screen. `Llama-3.1-8B` (Instruct) and `Ministral-8B` (Instruct-2410) were served locally with the API-served customer and screened on the test candidates at seed 0. `Llama-3.1-8B` first sent list arguments as JSON-encoded strings, which the environment reads character by character (it solved 2 of 54 test candidates); with those strings decoded (Appendix A.2) it solved 0 of 30, mostly by acting before confirming with the customer or by transferring to a human. `Ministral-8B` solved 0 of 15, repeating calls the environment had rejected. With no solved task there is no unit to put a fault in, so the screens were stopped and neither family enters any analysis; the screens are in the archive. Two API-served families were also started and stopped: the `claude-sonnet-4-6` screen was stopped for cost after 23 of the 55 test candidates (over 30 times `gpt-4o-mini`'s cost per episode; its spend is not in Table 1), and `gemini-3-flash-preview` failed a two-task smoke test with tool-call errors. Neither enters any analysis, and neither partial run is in the archive.

The $j = 8$ lag. Run on two arms as exploratory: j=4 minus j=8 is -0.024 $[-0.167, +0.118]$ on `gpt-4o-mini` ($n = 41$ of 56 units) and $+0.000$ on `gpt-5-mini` ($n = 10$ of 28). Both policies recover almost nothing by $j = 4$ already, so this shows a floor, not saturation, and says nothing about P4 where recovery remains. Only episodes still running at turn 8 enter, so the cell is a selected subsample, and both arms are already near the never-corrected floor at $j = 4$; it is excluded from every slope fit.