CONTEXT ROT IN AGENT TRAJECTORIES: EVOLVING-STATE MAINTENANCE IS THE HIDDEN BOTTLENECK OF LONG-CONTEXT AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-context language models are increasingly deployed as agents whose context is not a static document but a *self-generated* multi-turn trajectory of (thought, action, observation) steps. We ask whether early key information is faithfully used as this trajectory grows. We introduce a controlled probe suite that holds task difficulty fixed while varying only trajectory length $K \in \{1, \ldots, 128\}$, across four task families that separate *passive recall* (key–value lookup, early constraints, multi-hop) from *active evolving-state maintenance*. Evaluating nine frontier models, we find a sharp dissociation: passive recall is largely robust (constraint and multi-hop stay at ceiling; KV recall degrades mildly), whereas evolving-state maintenance collapses catastrophically for *every* model once $K \geq 16$ (early-information utilization drops from ≈ 1.0 to ≈ 0.0; all decay slopes $\lambda < 0$). We further isolate an agent-specific corruption source: a model's own erroneous intermediate conclusion written into the trajectory (*self-conflict*) degrades performance significantly more than neutral filler of equal length ($\Delta = -0.150$, 95% CI $[-0.199, -0.102]$); a controlled comparison shows this is driven by the misleading *content* rather than self-authorship per se, except for the strongest reasoner. Moreover 47.5% of these failures land exactly on the "snowball" value the model would obtain by faithfully continuing from its own error—an effect that *worsens with reasoning strength*. A two-phase mechanism emerges: shallow-K failures are dominated by self-generated-error snowballing (which a from-scratch recompute prompt partially repairs for weaker models, e.g. $+0.333$ for Claude Haiku), while deep-K failures are dominated by forgetting (which no training-free prompt repairs; a mechanism-targeted recompute recovers $+0.211$, 95% CI $[+0.139, +0.289]$, in the shallow snowball regime but $+0.007$, not significant, at the deep cliff). We corroborate both the cliff and the self-error effect on *genuine multi-turn rollouts* in which each model self-generates its own trajectory: the collapse to EUR ≈ 0 reproduces, and rollouts containing a self-authored intermediate error fail $6.1\times$ more often. Together these establish evolving-state maintenance as a stubborn, agent-specific long-context bottleneck and provide a first partial solution scoped to where it works.

1 INTRODUCTION

Frontier language models are increasingly deployed as *agents*: rather than answering a single query, they interleave reasoning, tool calls, and observations over many turns, accumulating a long context that they themselves generate. A growing body of work studies how models use long *static* contexts—documents, retrieved passages, or synthetic haystacks—and documents phenomena such as the "lost in the middle" positional bias (Liu et al., 2024a), degradation beyond literal matching (Modarressi et al., 2025), and failure to notice absent information (Fu et al., 2025). Yet the context of a deployed agent differs in kind from a static document: it is a *self-generated trajectory* of (thought, action, observation) steps, in which the model's own past outputs—including its mistakes—become part of the context that conditions its future behavior.

We ask a simple question: **as an agent's trajectory grows longer, does it still faithfully use the key information established early in that trajectory?** Figure 1 previews the two phenomena

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

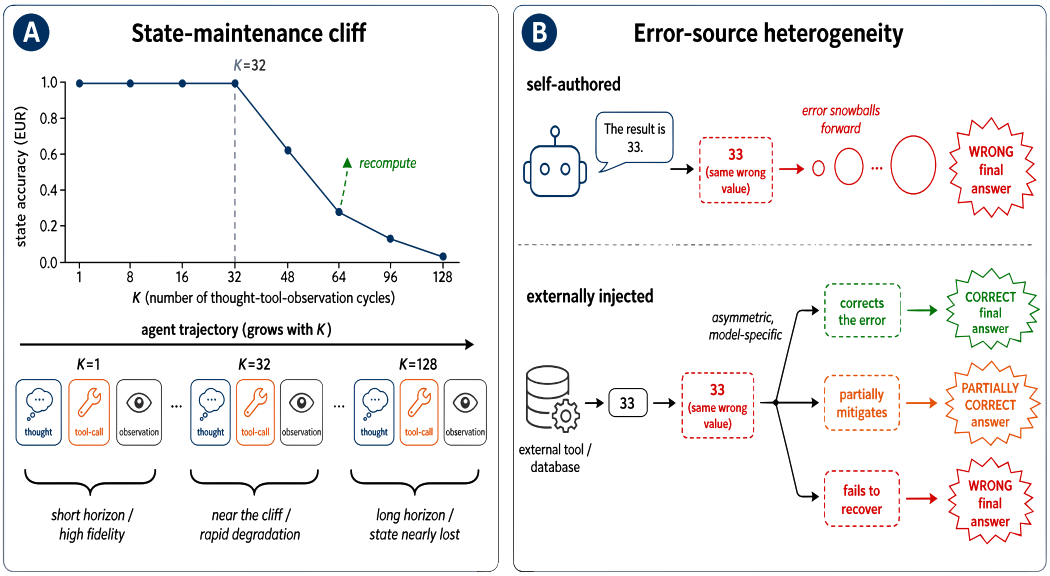


Figure 1: **Two phenomena in self-generated agent trajectories.** *(1) State-maintenance cliff.* As the trajectory length grows ($K{=}1 \rightarrow 128$ interleaved thought/tool-call/observation steps), a model's ability to maintain evolving state stays high, then collapses sharply once $K{\geq}32$; a lightweight *recompute* step can lift it back up. *(2) Error-source heterogeneity.* A wrong value that is *self-authored* (the model itself emits it) snowballs forward and corrupts the final answer, whereas the *same* wrong value *externally injected* by a tool/environment triggers a different, model-dependent internal reaction that may correct, partially mitigate, or fail to recover—self-error versus external-error is an asymmetric, model-specific vulnerability.

we uncover. Answering this cleanly requires disentangling trajectory *length* from task *difficulty*. We therefore design a controlled probe suite in which the reasoning complexity needed to answer is held fixed while only the number of intervening trajectory steps $K \in \{1, 2, 4, 8, 16, 32, 64, 128\}$ varies. Our central diagnostic is **early-information utilization** $\textsc{eur}(K)$: the rate at which the model correctly uses information placed at the start of a K-step trajectory, and its log-scale decay slope λ.

Crucially, we separate two kinds of demand on the early information. Three task families require only *passive recall*—the early fact sits statically in the context and must merely be retrieved at the end (key–value lookup, an early formatting constraint, and multi-hop clue aggregation). A fourth family, STATE-TRACK, requires *active evolving-state maintenance*: the model sets an initial counter and must integrate a $\pm$ event at *every* step to report the current value at the end. This distinction turns out to be decisive.

Findings. Across nine frontier models we observe a sharp dissociation (Section 4). Passive-recall tasks are largely robust: constraint-following and multi-hop remain at ceiling out to $K = 128$, and KV recall degrades only mildly. In stark contrast, evolving-state maintenance *collapses for every model* once $K \geq 16$—utilization falls from ≈ 1.0 at $K{=}1$ to ≈ 0.0 at $K{=}128$, with uniformly negative decay slopes. Reasoning models delay but do not escape the collapse. This updates a natural intuition (that "counting"-style accumulation is easy) and locates the real long-context bottleneck for agents in *maintaining a state that evolves with the interaction*.

We then isolate what makes agent trajectories distinctively fragile (Section 5). A *self-conflict* step—the model's own erroneous intermediate conclusion carried forward in the trajectory—degrades performance significantly more than neutral filler of equal length ($\Delta = -0.150$, 95% CI $[-0.199, -0.102]$; largest at mid-depth, $\Delta = -0.432$ at $K{=}4$). A controlled comparison that holds the misleading content identical and varies only its *source* reveals a counter-intuitive asymmetry: a wrong running total *reported by a tool* is far more damaging than the *same* value stated by the agent itself (self-conflict vs. external, item-level EUR over nine models: $+0.096$, 95% CI $[+0.068, +0.125]$; up to $+0.271$ at short trajectories), overturning the naive "models blindly trust

their own output" hypothesis and pointing instead to *source-authority anchoring*—tool feedback is the most blindly-trusted, most dangerous corruption entry point. Reconstructing what the model *would* compute by faithfully continuing from its own error, we find 47.5% of self-conflict failures land within ± 2 of this "snowball" value—and the effect *intensifies with reasoning strength* (69% for a reasoning model), a counter-intuitive liability: stronger reasoning more faithfully propagates a self-generated error. A two-phase picture emerges: shallow-K corruption is error *snowballing* (80% of failures at $K{=}1$), while deep-K corruption is *forgetting* (22% snowball at $K{=}128$). Both the cliff and the self-error effect reproduce on *genuine multi-turn rollouts* in which each model self-generates its trajectory via a tool-use loop.

Finally, we ask whether the collapse can be repaired without training (Section 6). Three prompt-level strategies leave the deep-K cliff ($K \geq 64$) essentially untouched for all models, but a paired analysis reveals a first partial solution in the mid-depth, snowball-dominated regime: *recompute* significantly helps weaker models, consistent with the mechanism—recomputing from scratch discards the running total that carries the model's own error.

Contributions.

- A *controlled* probe suite that holds task difficulty fixed and varies only self-generated trajectory length, separating passive recall from active evolving-state maintenance (Section 3).
- A large-scale evaluation of nine frontier models showing evolving-state maintenance is a *universal* long-context bottleneck that collapses beyond $K{=}16$, while passive recall is comparatively robust (Section 4).
- Identification of an *agent-specific* corruption source—the model's own erroneous intermediate output—with a two-phase (snowball then forgetting) mechanism, including the counter-intuitive finding that reasoning strength amplifies self-error propagation (Section 5).
- An honest evaluation of training-free mitigations: all fail on the deep-K cliff, but *recompute* is a statistically significant first partial solution in the mid-depth regime the mechanism predicts (Section 6).

2 Related Work

Long-context utilization in static settings. A large literature probes how models use long *static* contexts: a positional "lost in the middle" bias (Liu et al., 2024a), degradation persisting once surface-form matching is removed (Modarressi et al., 2025), and difficulty reporting what is *missing* (Fu et al., 2025). Synthetic stress tests (Hsieh et al., 2024; Li et al., 2024a) and broad suites (Bai et al., 2024; Zhang et al., 2024b; Yen et al., 2025) standardize evaluation, and long in-context learning remains hard (Li et al., 2024b). Most directly, Levy et al. (2024) isolate input length by padding a fixed task and observe reasoning decay before the nominal limit—industry reports term this "context rot" (Hong et al., 2025). All place the target in a *fixed, externally authored* document; none study a context the model itself generates turn by turn, nor a state that evolves with the interaction. Our probes hold difficulty fixed and vary only self-generated trajectory length, isolating an axis these benchmarks do not.

Interactive agents and long-horizon memory. Agent frameworks interleave reasoning and acting (Yao et al., 2023) and manage context to escape window limits (Packer et al., 2023; Zhang et al., 2024c). Agentic evaluations (Liu et al., 2024b; Zhou et al., 2024) report weak long-term reasoning as a dominant failure, and memory benchmarks (Maharana et al., 2024; Wu et al., 2025) emphasize *retrieval* of past user information over sessions. Our focus is complementary: not retrieval of a past fact, but continuous maintenance of a numeric state updated at every step, and the way a model's own past errors are inherited by its future context.

Error accumulation and self-generated corruption. Autoregressive generation is known to compound its own mistakes: exposure bias in sequence models (Zhang et al., 2019) and hallucination *snowballing* (Zhang et al., 2024a), where a model over-commits to an early error and then rationalizes further errors. Self-consistency mitigates single-path errors by sampling and voting (Wang

et al., 2023). We show that in long agent trajectories this dynamic becomes a dominant failure mode—a model faithfully propagates its own earlier miscalculation—and that our *self-conflict* condition, in which the distracting content is the model's own erroneous intermediate conclusion, is strictly more damaging than an external distractor (Shi et al., 2023) matched in length and topic.

Source authority and knowledge conflict. When a model's internal belief clashes with contextual evidence, it exhibits a consistent trust bias: sycophancy toward user signals (Sharma et al., 2024), over-reliance on parametric memory (Longpre et al., 2021), and a measurable tug-of-war between prior and external evidence (Xie et al., 2024; Wu et al., 2024a). Prior work frames this conflict as *parametric vs. external*; we surface a distinct axis inside a single trajectory—*self-generated vs. external*—and find that a model anchors on its own prior output even when a fresh external observation contradicts it.

Controlled synthetic diagnosis. Isolating a single competence through synthetic tasks is a long tradition, from "toy" reasoning (Weston et al., 2016) to counterfactual variants separating genuine reasoning from recitation (Wu et al., 2024b) and multi-hop consistency after knowledge editing (Zhong et al., 2023). Following this paradigm, we build synthetic trajectories with a deterministically computed gold state and fixed difficulty, so trajectory length is the sole controlled variable—removing any dependence on an LLM judge. To our knowledge we are the first to (i) hold difficulty fixed while scaling self-generated trajectory length, (ii) show evolving-state maintenance—not passive recall—is the dominant long-context failure mode for agents, and (iii) attribute a substantial fraction of failures to a model faithfully propagating its own earlier mistake.

3 A Controlled Probe Suite for Agent Trajectories

Design principle: fix difficulty, vary only trajectory length. Our goal is to measure how faithfully a model uses *early* key information as its trajectory grows, *independent* of task difficulty. We therefore construct probes in which the reasoning needed to answer the final question is held constant—it always depends on a single early fact (or a fixed small set) plus the final question—while the number of intervening trajectory steps $K \in \{1, 2, 4, 8, 16, 32, 64, 128\}$ varies. The context is not an external document but a *self-generated agent trajectory*: each intervening step is a (thought, action, observation) record simulating a tool call and its result, so that longer K reproduces the way a deployed agent's history accumulates.

Four task families: passive recall vs. active maintenance. The key intended before the padding steps is placed early (step 1–2), followed by K padding steps, then the final question. Three families require only *passive recall*—the early fact sits statically in the trajectory and must merely be retrieved at the end; a fourth (D) requires *active evolving-state maintenance*:

- **A: trajectory-KV recall.** Step 1 injects a key–value fact (e.g. "deployment port of project X = 8788"); K padding steps query unrelated tools; the final question asks for the value.
- **B: early-constraint following.** An early instruction imposes a constraint (e.g. "report all amounts in EUR"); after K unrelated subtasks, a question requiring an amount tests whether the constraint is still obeyed.
- **C: early-multihop aggregation.** Two–three early steps inject clues that must be chained ($A \to B$, $B \to C$); after K padding steps the model must bridge $A \to C$.
- **D: state-tracking (STATE-TRACK).** Step 1 sets an initial counter (e.g. "active sessions = 37"); *each* of the K padding steps reports a $\pm$ event that the model must integrate into a running total; the final question asks for the current value. Unlike A–C, the target is not a static fact to retrieve but a state that must be maintained across *every* step. The gold value is computed exactly by the generator from the initial value and all true deltas (validated with zero mismatch).

This passive-vs-active split is the crux of our study: our central hypothesis is that passive recall (A–C) is flat in K whereas evolving-state maintenance (D) decays sharply.

Table 1: Decay slope λ (EUR vs. $\log_2 K$) by model and task. B/C (passive recall) are flat; D (evolving-state maintenance) collapses for every model. More negative = faster context rot.

Model	A: KV	B: constraint	C: multihop	**D: state-track**
Claude Haiku 4.5	$\sim$	0.000	0.000	$\mathbf{-0.161}$
Claude Sonnet 4.6	$\sim$	0.000	0.000	$\mathbf{-0.180}$
DeepSeek-R1	-0.060	0.000	0.000	$\mathbf{-0.106}$
DeepSeek-V3.2	$\sim$	0.000	0.000	$\mathbf{-0.152}$
Gemini-2.5-Flash	$\sim$	0.000	0.000	$\mathbf{-0.094}$
Gemini-3.5-Flash	$\sim$	0.000	0.000	$\mathbf{-0.085}$
GPT-4.1	-0.016	0.000	0.000	$\mathbf{-0.154}$
GPT-5.1	$\sim$	0.000	0.000	$\mathbf{-0.164}$
Qwen3-235B	$\sim$	0.000	0.000	$\mathbf{-0.153}$

Three noise conditions. The padding steps come in three matched types of increasing agent-specificity: (i) **irrelevant**—neutral tool outputs unrelated to the target (pure length filler); (ii) **distractor**—topic-related but non-answer content (for STATE-TRACK, $\pm$ events on *other* counters explicitly marked as not affecting the target, tempting spurious updates); and (iii) **self-conflict**—the agent's own erroneous intermediate conclusion inserted into the trajectory (for STATE-TRACK, a wrong running total the model "reports" mid-way). Only self-conflict can arise from the model itself, and we match all three in length and, for the paired analysis of Section 5, in event stream and error content.

Metrics. Our primary metric is **early-information utilization** EUR(K): the fraction of instances at trajectory length K for which the early key information is correctly used (exact normalized match for KV/constraint/multihop and the numeric STATE-TRACK answer; an LLM judge with recorded version resolves ambiguous cases). We summarize each model–task by the **decay slope** λ, the regression coefficient of EUR on $\log_2 K$ (more negative = faster rot). For diagnosis we decompose each error into *forget* (never mentions the early information), *distracted* (answers with distractor content), *self-misled* (answers with the model's own wrong intermediate conclusion), and *early-stop* (uninformative answer / refusal).

Protocol. Probes are stored as `data/probes/<task>_<noise>_K{K}.jsonl`. We use incremental, sample-level checkpointing (only successful samples are written and flushed), enabling seamless resume, and run each model in an independent process for throughput. Unless stated otherwise we report EUR over n=15 instances per cell with k=3 majority-vote samples.

4 Evolving-State Maintenance Is the Long-Context Bottleneck

We evaluate nine frontier models spanning four vendors and both standard and reasoning variants: GPT-4.1, GPT-5.1, Claude Sonnet 4.6, Claude Haiku 4.5, DeepSeek-V3.2, DeepSeek-R1, Qwen3-235B, Gemini-2.5-Flash, and Gemini-3.5-Flash. For each model we measure EUR(K) on all four task families under the irrelevant condition (the cleanest length-only manipulation), and fit the decay slope λ.

A sharp dissociation between recall and maintenance. Table 1 reports λ per model and task, and Figure 2 shows the EUR(K) curves. The pattern is unambiguous and consistent across all nine models: *passive-recall tasks resist trajectory length, whereas evolving-state maintenance collapses.* Constraint following (B) and multi-hop aggregation (C) stay at ceiling ($\lambda \approx 0.000$) out to K=128 for essentially every model; KV recall (A) degrades only mildly (e.g. $\lambda_{\text{GPT-4.1}} = -0.016$, $\lambda_{\text{R1}} = -0.060$). In stark contrast, state-tracking (D) collapses for *every* model: EUR falls from ≈ 1.0 at K=1 to ≈ 0.0 at K=128, with uniformly steep negative slopes.

The collapse is universal and begins at K=16. Figure 3 shows the state-tracking EUR(K) curve for each model. Every model starts near ceiling at short trajectories ($K \leq 4$) and falls to the floor by K=128. The onset is remarkably consistent: all nine models are already degrading by K=16 and essentially collapsed by K=64. Representative endpoints (K=1 $\rightarrow$ K=128): Claude Sonnet

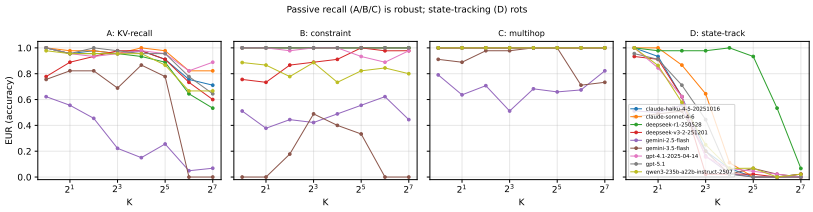


Figure 2: EUR(K) by task family, averaged over nine models. Passive-recall tasks (A/B/C) remain near ceiling as trajectory length grows; evolving-state maintenance (D) collapses beyond K=16.

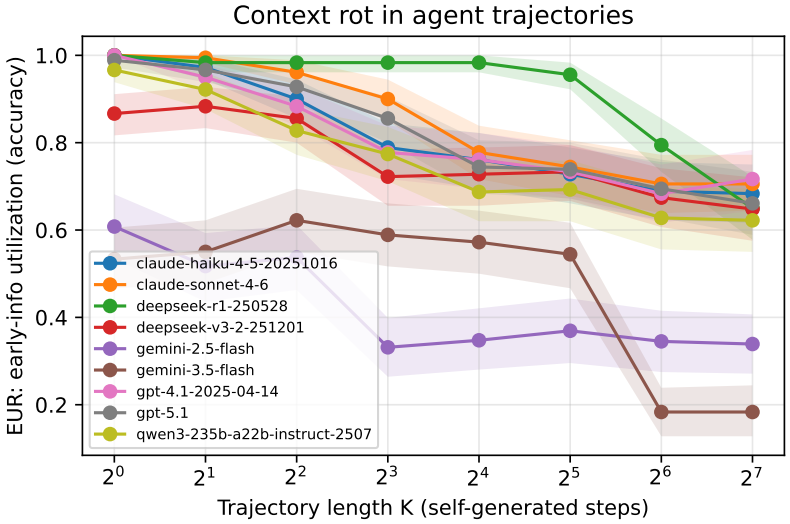


Figure 3: State-tracking EUR(K) for each of the nine models. Every model starts near ceiling at short trajectories and collapses to the floor by K=128; degradation is already visible by K=16. Reasoning models delay the cliff but do not escape it.

1.00 → 0.00, GPT-4.1 1.00 → 0.00, GPT-5.1 0.96 → 0.00, Qwen3-235B 1.00 → 0.02, DeepSeek-V3.2 0.93 → 0.02, Claude Haiku 1.00 → 0.02.

Reasoning delays but does not prevent collapse. The most robust model is the reasoning model DeepSeek-R1 ($\lambda = -0.106$; still 0.53 at K=64 before falling to 0.07 at K=128), indicating that explicit reasoning *delays* the cliff but does not remove it. Notably, GPT-5.1—also a strong reasoning model—collapses as steeply as standard models ($\lambda = -0.164$), so reasoning capability alone is not protective; we return to why in Section 5, where stronger reasoning is shown to *amplify* a specific self-error failure mode.

The collapse reproduces on genuinely self-generated trajectories. Our main suite uses controlled, templated trajectories to isolate the length axis. To confirm this is not an artifact of that construction, we run a *genuine multi-turn rollout* subset in which the agent itself drives the trajectory: each turn it calls a `next_event` tool, reasons, updates a running total, and reports the final value only at the end; the trajectory is entirely self-generated and the gold value is computed exactly from the true event stream. We roll out eight of the nine models across the full $K \in \{1, 2, 4, 8, 16, 32, 64, 128\}$ grid and the three noise conditions (22,317 scored rollouts; Claude Sonnet 4.6 is excluded here as it hallucinates `next_event` calls beyond the given stream, an agentic artifact orthogonal to length rot—its templated-suite results are unaffected; a diagnostic re-run with a prompt that forbids inventing events restores its normal profile, near-ceiling EUR through K=32 followed by the same deep-K collapse, confirming the exclusion targets the arti-

fact rather than the model). The K-cliff reproduces cleanly: aggregate EUR falls monotonically from 0.97 at $K{=}1$ to 0.22 at $K{=}128$ (supplement), matching the templated suite. Reasoning models again degrade more slowly (DeepSeek-R1 holds EUR$=0.63$ at $K{=}128$, GPT-5.1 near ceiling at 0.99) while lightweight models collapse earliest, reproducing the same model ordering as the controlled probes. Crucially, the self-error effect also reproduces *causally* on real trajectories: we log, per rollout, whether the agent ever writes a running total that deviates from ground truth (a genuine, self-authored *mid-trajectory conflict*). Rollouts that contain such a self-generated error fail 81% of the time ($3{,}322/4{,}122$), versus 13% for rollouts without one ($2{,}393/18{,}195$)—a $6.1\times$ increase in failure rate. On authentic self-generated context, an agent that commits an intermediate error almost always carries it through to a wrong final answer, exactly the snowball mechanism dissected in Section 5.

Takeaway. The long-context bottleneck for agents is not passive recall—comparatively robust—but *active maintenance of a state that evolves with the interaction*, which collapses universally beyond a short horizon in both controlled probes and genuine rollouts, motivating the mechanistic and mitigation analyses that follow.

5 An Agent-Specific Corruption Source and Its Mechanism

Section 4 establishes *that* evolving-state maintenance collapses. We now ask *why* agent trajectories are distinctively fragile, focusing on state-tracking (STATE-TRACK) where the collapse is sharpest.

5.1 What makes self-generated conflict damaging: content, not authorship per se

We first compare the three noise conditions at matched trajectory length. A *self-conflict* step—where the model's own erroneous running total is carried forward—is significantly more damaging than a *neutral* (irrelevant) filler step: aggregating the paired difference EUR(self-conflict) $-$ EUR(irrelevant) over all 72 model$\times K$ cells yields $\Delta = -0.150$, 95% CI $[-0.199, -0.102]$ (paired bootstrap), significant overall. The gap is concentrated at mid-depth, where the model has had time to commit an error but not yet forgotten everything: $\Delta = -0.432$ at $K{=}4$ (worse in $9/9$ models) and $\Delta = -0.299$ at $K{=}8$ ($9/9$). This specificity is visualized in the supplementary material. We do *not* claim a steeper slope—self-conflict's λ is in fact the flattest (-0.110), because it starts lower with less room to fall; the effect is on *absolute* utilization at matched length.

Does the *source* of the wrong state matter? A natural hypothesis is that self-*authored* content is uniquely corrupting (the model blindly trusts its own past output). We test this directly with a controlled paired comparison that holds the misleading content *identical* and varies only who produced it: the same wrong running total is either (i) stated by the agent itself (self-conflict) or (ii) injected verbatim by an external tool observation (*external*). Event stream, error value, and position are matched per-instance, so the sole variable is the *source* of the error. Across all nine models, using our item-level EUR metric, the paired difference EUR(self-conflict) $-$ EUR(external) is $+0.096$, 95% CI $[+0.068, +0.125]$: on balance the agent is *more* accurate when the wrong total is self-stated than when a tool reports it. The effect is strongest at short trajectories—$+0.271$ ($[+0.173, +0.376]$) at $K{=}1$ and $+0.241$ ($[+0.128, +0.353]$) at $K{=}2$—and remains significant through the mid-to-deep range ($K{\in}\{16, 32, 64\}$: $+0.090, +0.104, +0.060$, all CIs excluding zero), before both sources collapse to the floor at $K{=}128$ (Fig. 4). This *overturns* the naive self-trust hypothesis: the more dangerous corruption is an *externally* reported wrong state, not a self-authored one. We attribute this to *source-authority anchoring*—a tool observation carries higher perceived authority than the agent's own passing remark, so a wrong tool value is adopted as ground truth while a self-stated one retains some skepticism. The agent-specificity is thus not that models trust *themselves*, but that agentic trajectories inject authoritative-looking tool outputs—the most blindly-trusted, most dangerous corruption entry point. At deep K this distinction vanishes because forgetting (Phase 2), not misplaced trust, dominates.

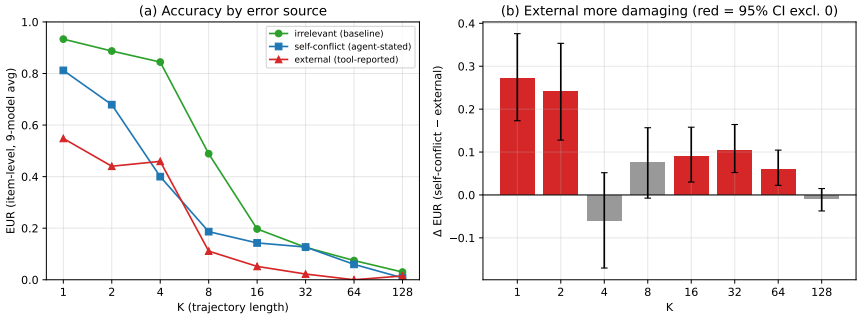


Figure 4: **Source-authority anchoring.** (a) Item-level EUR (9-model average) under three matched conditions: an irrelevant baseline, a *self-conflict* wrong running total stated by the agent, and an *external* wrong total reported by a tool observation. The externally reported error is consistently the most damaging until deep K, where forgetting flattens all sources. (b) Paired ΔEUR(self-conflict $-$ external) by K; red bars have 95% bootstrap CIs excluding zero. The gap is largest at short trajectories and persists through mid-to-deep K.

5.2 Failures land on the model's own "snowball" value

To test whether the model is literally propagating its own error, we reconstruct the *snowball value*: the total the model would obtain by faithfully continuing arithmetic from its injected wrong intermediate conclusion. We then ask what fraction of self-conflict failures land within ± 2 of this snowball value. Averaged across models, 47.5% of failures are snowball hits—far above chance for an unconstrained integer answer—directly implicating self-error propagation rather than random corruption.

Reasoning strength amplifies self-error propagation. Counter-intuitively, the snowball effect is *stronger* for more capable reasoners: the reasoning model exhibits a 69% snowball-hit rate, versus lower rates for weaker models. Stronger step-by-step reasoning more faithfully carries a self-generated premise to its (wrong) conclusion—a liability rather than a safeguard when the premise is the model's own earlier mistake.

5.3 A two-phase mechanism: snowball then forgetting

The snowball-hit rate is not constant in K; it reveals two regimes (visualized in the supplementary material). At shallow trajectories, failures are dominated by error snowballing: 80.0% of self-conflict failures at $K{=}1$ are snowball hits (82.6% at $K{=}2$, 73.5% at $K{=}4$). As K grows this fraction decays monotonically—56.5% at $K{=}16$, 37.0% at $K{=}32$, 21.9% at $K{=}128$—as a second, length-driven mechanism takes over: *forgetting*. The failure-mode decomposition confirms this shift: at $K{=}1$ the dominant modes are *self-misled* and *early-stop*, whereas at $K{=}128$ *forget* and *distracted* dominate (all four modes are large at long K; see the supplementary material). Thus the collapse is not monolithic: shallow-K corruption is the agent propagating its own error; deep-K corruption is the early state being forgotten.

Human validation of the failure attribution. Because the failure-mode decomposition above rests on an automatic attribution heuristic, we validate it against human judgment. Five independent annotators each labeled the same 100 failure cases (stratified over models, K, and noise) into five categories—snowball, forget, distract, arithmetic, and other—following a written guideline. Inter-annotator agreement is *substantial*: Fleiss' $\kappa = 0.58$ across all five raters (mean pairwise Cohen's $\kappa = 0.58$), and a majority-vote consensus label exists for all 100 items (48 unanimous, 91 with a strict majority). The consensus distribution is dominated by *snowball* ($45/100$) and *arithmetic* ($30/100$), consistent with the automatic decomposition and confirming that self-error propagation is the single largest human-adjudicated failure class. This human-validated set anchors the mechanistic claims to reliable labels rather than to the heuristic alone.

This mechanism makes a concrete prediction for mitigation: an intervention that *discards* the running total (and hence the model's own error) should help in the shallow, snowball-dominated regime, but nothing prompt-level should repair deep-K forgetting. We test exactly this next.

6 Mitigation: Where Training-Free Prompting Helps and Where It Fails

The mechanism of Section 5 predicts a narrow window in which a training-free intervention could help: the shallow, snowball-dominated regime, where discarding the running total removes the model's own error. We evaluate three prompt-level strategies against a no-intervention baseline on state-tracking: (i) **scratchpad**—maintain an explicit externalized tally; (ii) **restate**—restate the current state before each action; and (iii) **recompute**—recompute the total from scratch from the original events rather than trusting the running total.

No strategy repairs the deep-K cliff. On the deep-K region ($K \geq 32$), all three strategies leave the collapse untouched ($\Delta \leq +0.01$ in EUR). This is the expected floor effect: once the early state is forgotten, no prompting recovers information that is no longer tracked—evidence that deep-K context rot is a stubborn, open problem.

Recompute is a significant first partial solution in the snowball regime. We test anchor-recompute—re-deriving the total from the original event anchors instead of the running tally—against the no-intervention baseline with paired bootstrap CIs (full results in the appendix). In the deep-K cliff ($K \geq 32$) pooled recovery is $+0.007$, 95% CI $[-0.015, +0.037]$—not significant. In the shallow, snowball-dominated regime ($K \leq 8$) pooled recovery is $+0.211$, $[+0.139, +0.289]$—statistically significant, and largest for the most snowball-driven model, GPT-5.1 ($+0.400$; K=4 recovery $+0.80$). Per-model, recompute significantly helps weaker models (Claude Haiku $\Delta = +0.333$, $[+0.239, +0.428]$; GPT-4.1 $+0.100$, $[+0.006, +0.189]$) while the strongest models, already high in this regime, are unaffected. Recompute works by discarding the running total that carries the model's own error, neutralizing the snowball while leaving the forgetting-driven cliff untouched—a tight mechanism–intervention loop and a first partial solution.

7 Discussion, Limitations, and Conclusion

Summary. Across nine frontier models, our probe suite reveals a sharp dissociation: passive recall of early information is robust, but *active maintenance of an evolving state* collapses universally beyond a short horizon. We identified an agent-specific corruption source—the model's own erroneous intermediate conclusion—and traced a two-phase mechanism (self-error snowballing at shallow K, forgetting at deep K), including the finding that reasoning strength amplifies self-error propagation. Training-free prompting cannot repair the deep-K cliff, but recompute is a significant first partial solution in the snowball regime.

Limitations. Our main probes use synthetic trajectories to isolate the length axis; we validate the core findings on genuinely self-generated rollouts (Section 4), where both the K-cliff and self-error effect reproduce. We do *not* claim a self-generated trajectory is strictly worse than an equal-token static document as a general law—our controlled static comparison is weak, so we leave the "rot premium" open. Our mitigation covers only prompt-level strategies; architectural or training-time interventions are a natural next direction.

Implications and conclusion. Practitioners should not trust a running state carried implicitly in the trajectory: periodically recomputing critical state from ground truth mitigates the shallow-K snowball, while deep-horizon maintenance likely requires external memory or structural changes rather than better prompting. For researchers, evolving-state maintenance under self-generated context is a concrete, measurable, currently unsolved challenge that static benchmarks miss. Context rot in agents is thus not primarily about forgetting an early fact—it is about failing to maintain an evolving state and inheriting one's own past mistakes. By isolating these two phenomena and showing a mechanism-aligned intervention partially addresses one, we provide both a diagnostic tool and a target for future long-context agent research.

Ethics Statement

This work introduces a controlled diagnostic benchmark for long-context language agents. All probe instances are synthetically constructed and all agent trajectories are generated in a fully simulated tool-use environment; no human-subject data, personal data, or private information is collected or released. The small human-validation study (Section 4) uses five volunteer annotators who labelled model failure cases only; no personal information about the annotators is recorded, and participation was voluntary. We foresee no direct harmful application: the benchmark is intended to surface, not to exploit, a robustness limitation of long-context agents, and the mechanism-targeted mitigation we study is a defensive prompting technique.

Reproducibility Statement

All results are reproducible from released artefacts. The probe generator, the rule-based scoring and failure-mode attribution, the paired-bootstrap and snowball-mechanism analysis scripts, and the mitigation runner are provided in the code supplement; fixed seeds are used throughout so that each self-conflict instance is paired to its matched control by construction, and every paired bootstrap uses a fixed seed. The evaluated synthetic-probe results, the 22,317 genuine multi-turn rollouts, the gold states, the five annotator label files, and all aggregated result files are released in the data supplement. Section 3 defines every task family and failure mode precisely, and the supplementary material gives the full annotation protocol and per-K breakdowns. Gold answers are computed exactly from the event stream rather than judged by a model, so scoring is deterministic and re-runnable.

Use of Large Language Models

Large language models are the object of study in this paper: the nine frontier models we evaluate are listed in Section 4. We do *not* use any LLM as a judge—every trajectory is scored deterministically against a programmatically computed gold state. No LLM was used to generate the scientific claims, experimental design, or analysis of this paper; any use of LLMs for language polishing did not affect the technical content.

References

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A bilingual, multitask benchmark for long context understanding. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

Harvey Yiyun Fu, Aryan Shrivastava, Jared Moore, Peter West, Chenhao Tan, and Ari Holtzman. AbsenceBench: Language models can't tell what's missing. *arXiv preprint arXiv:2506.11440*, 2025.

Kelly Hong, Anton Troynikov, and Jeff Huber. Context rot: How increasing input tokens impacts LLM performance. Technical report, Chroma, 2025. URL `https://research.trychroma.com/context-rot`. Chroma Research.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? In *Conference on Language Modeling (COLM)*, 2024.

Mosh Levy, Alon Jacoby, and Yoav Goldberg. Same task, more tokens: the impact of input length on the reasoning performance of large language models. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

Mo Li, Songyang Zhang, Taolin Zhang, Haodong Duan, Yunxin Liu, and Kai Chen. NeedleBench: Evaluating LLM retrieval and reasoning across varying information densities. *arXiv preprint arXiv:2407.11963*, 2024a.

Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. Long-context LLMs struggle with long in-context learning. *arXiv preprint arXiv:2404.02060*, 2024b.

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*, 2024a.

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

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

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of LLM agents. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

Ali Modarressi, Hanieh Deilamsalehy, Franck Dernoncourt, Trung Bui, Ryan A. Rossi, Seunghyun Yoon, and Hinrich Schütze. NoLiMa: Long-context evaluation beyond literal matching. *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. MemGPT: Towards LLMs as operating systems. *arXiv preprint arXiv:2310.08560*, 2023.

Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R. Bowman, Newton Cheng, Esin Durmus, Zac Hatfield-Dodds, Scott R. Johnston, et al. Towards understanding sycophancy in language models. In *International Conference on Learning Representations (ICLR)*, 2024.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H. Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M. Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. Towards ai-complete question answering: A set of prerequisite toy tasks. In *International Conference on Learning Representations (ICLR)*, 2016.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. LongMemEval: Benchmarking chat assistants on long-term interactive memory. In *International Conference on Learning Representations (ICLR)*, 2025.

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 (NeurIPS) Datasets and Benchmarks Track*, 2024a.

Zhaofeng Wu, Linlu Qiu, Alexis Ross, Ekin Akyürek, Boyuan Chen, Bailin Wang, Najoung Kim, Jacob Andreas, and Yoon Kim. Reasoning or reciting? exploring the capabilities and limitations of language models through counterfactual tasks. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2024b.

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 (ICLR)*, 2024.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Howard Yen, Tianyu Gao, Minmin Hou, Ke Ding, Daniel Fleischer, Peter Izsak, Moshe Wasserblat, and Danqi Chen. HELMET: How to evaluate long-context language models effectively and thoroughly. In *International Conference on Learning Representations (ICLR)*, 2025.

Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A. Smith. How language model hallucinations can snowball. In *International Conference on Machine Learning (ICML)*, 2024a.

Wen Zhang, Yang Feng, Fandong Meng, Di You, and Qun Liu. Bridging the gap between training and inference for neural machine translation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Khai Hao, Xu Han, Zhen Leng Thai, Shuo Wang, Zhiyuan Liu, and Maosong Sun. ∞bench: Extending long context evaluation beyond 100k tokens. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024b.

Zeyu Zhang, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Quanyu Dai, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. A survey on the memory mechanism of large language model based agents. *arXiv preprint arXiv:2404.13501*, 2024c.

Zexuan Zhong, Zhengxuan Wu, Christopher D. Manning, Christopher Potts, and Danqi Chen. MQuAKE: Assessing knowledge editing in language models via multi-hop questions. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. WebArena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations (ICLR)*, 2024.

A Additional Mechanism Figures

This appendix collects supporting figures referenced in the main paper. They corroborate, at finer granularity, the mechanism results reported in the main text; the main-text figures already summarize the headline effects.

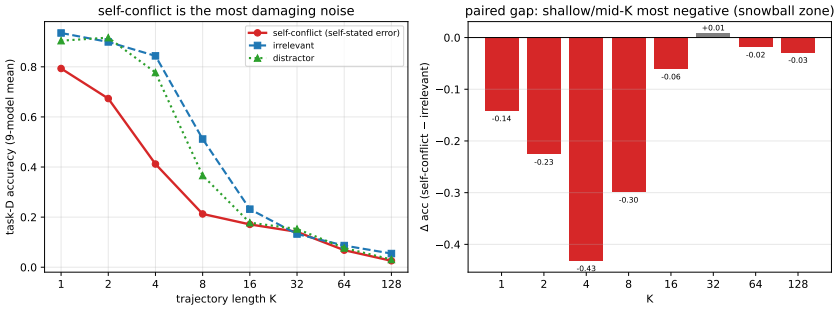


Figure 5: Self-conflict specificity on STATE-TRACK. At matched trajectory length, misleading content (a wrong running total) in the trajectory degrades EUR significantly more than neutral filler, peaking at mid-depth (K=4–8). A controlled comparison in the main text shows this is driven by content rather than authorship in general, with self-authorship an extra liability only for the strongest reasoner.

B Genuine Multi-Turn Rollouts: Per-Model Breakdown

Figure 8 shows the per-model EUR(K) curves on the genuine multi-turn rollout subset described in Section 4. The K-cliff reproduces the templated-suite finding under real generation for all eight models in this subset; see the main text for aggregate endpoints and the self-authored-error failure-rate contrast.

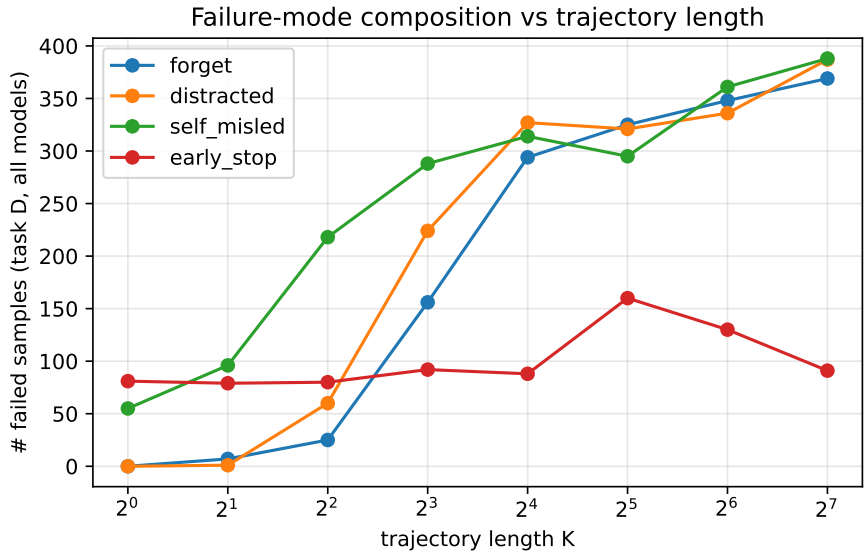


Figure 6: Failure-mode composition on STATE-TRACK by trajectory length. Self-misled and early-stop dominate at short K; forget and distracted dominate at long K, consistent with the two-phase mechanism described in the main text.

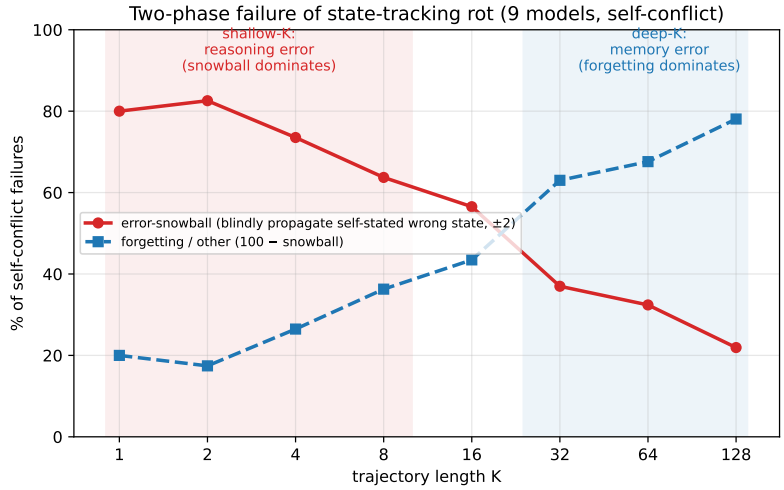


Figure 7: Two-phase mechanism. Snowball-hit rate (share of self-conflict failures matching the self-propagated value) is high at shallow K (error snowballing) and decays as trajectory length grows and forgetting dominates.

C MITIGATION: FULL PAIRED EFFECTS TABLE

Table 2 reports the full paired mitigation effects for all five models and three strategies in the mid-depth regime, complementing the pooled analysis in Section 6. Figure 9 visualizes the same results across all trajectory lengths. Only recompute yields statistically significant gains, and only for weaker models in the snowball-dominated regime.

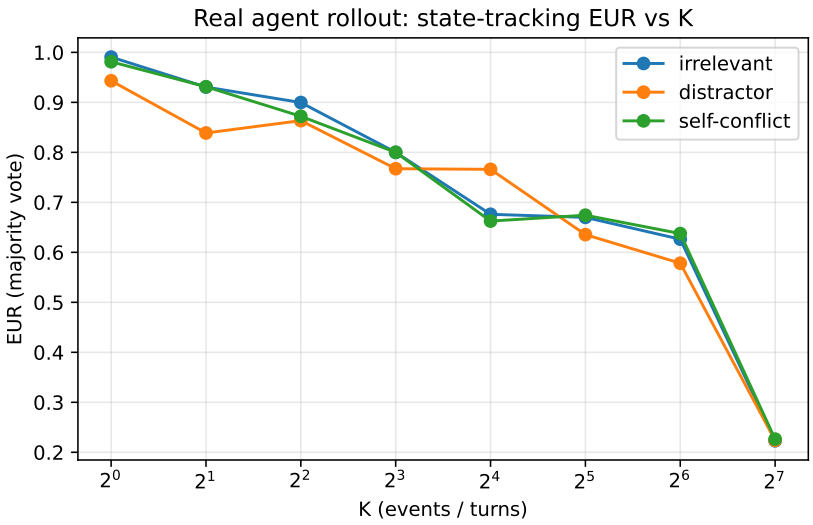


Figure 8: EUR(K) on *genuine multi-turn rollouts*, where each model self-generates its trajectory via a tool-use loop, aggregated over the eight models included in this subset (Claude Sonnet 4.6 excluded; see main text). The K-cliff reproduces the templated-suite finding under real generation; see the main text for the aggregate endpoints and the self-authored-error failure-rate contrast.

Table 2: Paired mitigation effect ΔEUR (strategy $-$ baseline) in the mid-depth regime, five models, with 95% bootstrap CIs. Only significant positive effects are shown; "·" denotes not significant. Recompute is a first partial solution for weaker models, matching the two-phase mechanism.

Model	recompute	restate	scratchpad
Claude Haiku 4.5	$+\mathbf{0.333}$ $[+0.239, +0.428]$	$+0.117$ $[+0.028, +0.206]$	·
GPT-4.1	$+0.100$ $[+0.006, +0.189]$	·	·
GPT-5.1	·	·	·
Qwen3-235B	·	·	·
DeepSeek-R1	·	·	·

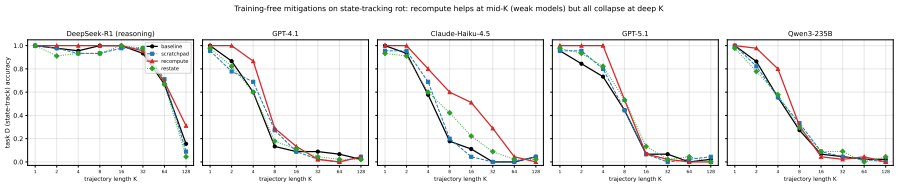


Figure 9: Mitigation effect by model and strategy. Recompute helps weaker models in the snowball-dominated regime; no strategy recovers the deep-K cliff.