Supersede: Diagnosing and Training the Memory-Update Gap in LLM Agents

Vedant Patel
Vrin · vedant@vrin.cloud

Code & Environment · Prime Intellect Hub · Model & Data

Abstract

Large language model (LLM) agents increasingly operate over long, multi-session interactions in which facts change: a user moves, a price updates, a plan is revised. Acting correctly requires using the *current* value of a fact and discarding values that have been *superseded*. We isolate this ability on real conversational data and show that it is a distinct, unsolved failure. On the `knowledge-update` subset of LongMemEval, replacing an agent's full context with a bounded, self-maintained memory drops accuracy from 92% to 77% even on a frontier model (gpt-5.4), a gap that is statistically significant (paired McNemar $p < 0.005$) and *persists across model scale* while full-context accuracy saturates near 92%. The bottleneck is therefore memory maintenance, not comprehension, and it is not closed by a stronger model. We then ask whether this is merely an artifact of an undersized memory, and find it is not: as the conversation grows 24×, accuracy falls further (from 68% to 28%), and granting the agent proportionally more memory yields no detectable recovery (28% → 28%, $n = 25$). The failure scales with the length of the conversation, not with the compression ratio. We release **Supersede**, an open reinforcement-learning environment (built on the `verifiers` / `prime-rl` stack) that turns this measurement into a training signal: agents are rewarded for answering from the current value and penalized for stale ones. Finally, we close the loop and show the gap is trainable: GRPO fine-tuning a small open model (Qwen2.5-3B) on this environment nearly doubles its held-out supersession accuracy on real, unseen conversations (9.0% → 16.7%, a single run), along a monotonic checkpoint curve that indicates the learned policy, not the harness, carries the gain. To our knowledge this is the first trainable environment whose reward targets temporal fact-currency, and the first evidence that the supersession gap can be trained down rather than only measured.

1 Introduction

Pre-training has largely consumed the open web, and the frontier of LLM capability has shifted toward agents that act over long horizons using *external memory* rather than a single context window. A defining property of long-horizon interaction is that information is not static: facts are introduced, then later updated or retracted. An assistant that is told a user lives in Detroit, and later that they moved to the suburbs, must answer a subsequent location question with the *current* value. We call the correct handling of such updates *supersession*, and the accuracy an agent loses when it must sustain it under a bounded, self-maintained memory the *supersession gap*. This paper measures that gap, shows the obvious remedies do not close it, and then trains it down.

Recent benchmarks have begun to measure long-term memory (Maharana et al., 2024; Wu et al., 2025; He et al., 2026; Hu et al., 2025; Uddin et al., 2026), and a forgetting-aware metric has been proposed to penalize reliance on obsolete information (Uddin et al., 2026). These works, however, score *frozen* models: they quantify the problem but do not provide

a mechanism to *train* it away. In parallel, a line of work applies reinforcement learning to memory agents (Yu et al., 2025; Chen et al., 2026), but rewards final-answer correctness or evidence relevance, never temporal currency. No existing work sits in the intersection: a *trainable* environment whose reward is supersession-correctness.

This paper makes five contributions, structured as one investigation: we ask in turn whether the failure exists, whether a bigger model fixes it, whether a bigger memory fixes it, and whether *training* fixes it — and answer no, no, no, and yes. The first three answers establish that the easy escapes are closed; the fourth shows the principled one is open.

1. **An isolation result.** On real conversational data, we hold memory load fixed and vary only whether the agent has full context or a bounded, self-maintained memory. Bounded memory degrades `knowledge-update` accuracy significantly (Section 5.1), isolating memory maintenance, not reading comprehension, as the cause.
2. **A frontier confirmation.** The gap survives on gpt-5.4 (92% $\rightarrow$ 77%, $p = 0.0033$) and does not shrink to zero as the model improves; full-context accuracy, by contrast, saturates. A bigger model does not fix it.
3. **Scale, not size.** As the conversation grows 24$\times$, the failure deepens (68% $\rightarrow$ 28%), and giving the agent proportionally more memory yields no detectable recovery at $n = 25$ (Section 5.2). The failure tracks the length of the conversation, not the compression ratio, so a bigger memory does not fix it either.
4. **An open environment.** We release **Supersede**[1], a `verifiers/prime-rl` environment with a programmatic supersession-aware reward, validated end-to-end. It reframes the forgetting-aware *metric* of Uddin et al. (2026) as a *learning signal*, so the failure can be trained against rather than only measured.
5. **Training closes it.** Fine-tuning a small open model (Qwen2.5-3B) on the environment with GRPO nearly doubles held-out supersession accuracy on real, unseen conversations (9.0% $\rightarrow$ 16.7%, Section 5.3; a single training run, multi-seed significance in ongoing work), monotonically as training progresses. Where a bigger model and a bigger memory did not, a trained memory policy does, turning the diagnosis into a fix.

2 Related Work

Long-term memory benchmarks. LoCoMo (Maharana et al., 2024) evaluates very long multi-session dialogue but does not annotate a dedicated update category. LongMemEval (Wu et al., 2025) introduces an explicit `knowledge-update` question type in which a fact stated in one session is changed in a later one; we adopt this subset as ground truth. MemoryArena (He et al., 2026) shows that models near-perfect on recall benchmarks drop sharply when memory must drive actions, motivating evaluation beyond passive recall. MemoryAgentBench (Hu et al., 2025) and MemBench (Tan et al., 2025) add knowledge-updating and conflict-resolution competencies. Memora and its FAMA metric (Uddin et al., 2026) explicitly penalize reliance on superseded or deleted memory, and temporal-conflict QA (Özer & Yıldız, 2025) probes whether models resolve facts that change over time. All of these are evaluation-only.

Memory systems and RL for memory. Mem0 (Chhikara et al., 2025) manages a memory store with prompted ADD/UPDATE/DELETE operations: a heuristic, not a learned, policy. A line of work instead *learns* memory management with RL: MemAgent (Yu et al., 2025) rewards final-answer correctness, LongRLVR (Chen et al., 2026) rewards evidence *relevance* within a single long document, AgeMem (Yu et al., 2026) learns store/update/discard

[1]Code, environment, trained model, and dataset are publicly available: https://github.com/Vrin-cloud/supersede (environment), https://huggingface.co/vedant33/supersede-qwen2.5-3b-grpo-lora (model), and https://huggingface.co/datasets/vedant33/supersede-rl-episodes (dataset); the environment is also published on the Prime Intellect Environments Hub.

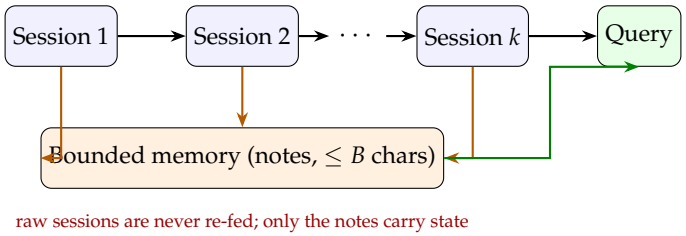

Figure 1: The Supersede rollout. The agent rewrites a bounded notes memory after each session and answers the final query from memory alone. Because history is not re-fed, a superseded value that is not overwritten persists and corrupts the answer.

operations under *task-outcome* rewards, and Memory-T1 (Du et al., 2025) learns to *select* temporally relevant sessions from the full history. Two things separate Supersede. First, these methods may re-read the raw history; Supersede forbids re-feeding, so the only state is the agent's own bounded memory. Second, their reward is task success, evidence relevance, or temporal consistency, not whether the agent kept the *current* value of a changed fact. Supersede makes that the reward itself: a verifiable, time-indexed supersession signal that reframes FAMA's forgetting-aware *objective* (Uddin et al., 2026) as a dense training target. The closest contemporary method, MemPO (Li et al., 2026), also pairs GRPO with a self-managed bounded memory, but rewards memory task-sufficiency (credit assignment over memory effectiveness), not which value is current. Put differently, prior RL-for-memory work learns *what to answer*; Supersede learns *which version is current*: the supersession signal is the reward, not a proxy for it.

Deployed memory systems. Memory reached production across every major lab in 2026: OpenAI's "Dreaming" (OpenAI, 2026), Anthropic's Claude memory (Anthropic, 2026), and Google's Gemini personalization (Google, 2026) all run background processes that consolidate and rewrite a user's memory to keep facts current. Tellingly, OpenAI's own time-sensitive evaluation of updating outdated context self-reports only 75.1% success (up from 9.4% in 2024), and Gemini's documentation notes that profile updates can lag by days. These systems are closed, with no released benchmark, methodology, or trainable signal: the gap Supersede fills.

Environment infrastructure. We build on the `verifiers` library and the Prime Intellect Environments Hub (Prime Intellect, 2025), which standardize environments for post-training, and are compatible with the emerging OpenEnv standard (Hugging Face and Meta PyTorch and contributors, 2026).

3 The Supersede Environment

Task. A task is a multi-session interaction followed by a query about the current value of a fact that changed during the interaction. The agent processes one session at a time and maintains a *bounded* memory (a notes field capped at B characters); crucially, raw sessions are *never re-fed*. After the final session, the agent answers the query using its memory alone. Figure 1 shows the rollout.

Reward. The primary reward is $r_{\text{cur}} = \mathbb{1}[\text{the final answer conveys the current value}]$, scored by a programmatic, ungameable matcher (normalized variant match with a token-overlap fallback), so the environment can be evaluated and trained without an auxiliary judge model. When the superseded values are known (our synthetic tasks, or update-annotated items), a penalty $r_{\text{stale}} = -\mathbb{1}[\text{the answer asserts a superseded value}]$ is added; on gold-only data it is inert. The combined reward is $r = r_{\text{cur}} + \lambda\, r_{\text{stale}}$. The runs reported here optimize r_{cur} alone (weight 1.0); the stale penalty enters only in the environment's ablation. This makes the forgetting-aware accuracy of Uddin et al. (2026) a dense training signal rather than a post-hoc score.

Data. Tasks are drawn from the LongMemEval `knowledge-update` subset (real conversational supersession) and from a procedural generator of synthetic timelines used for controls and ablations. The environment is implemented as a `verifiers` `MultiTurnEnv`; the prompt assembly enforces the no-re-feed property, and rollouts terminate when the answer is produced.

4 Experimental Setup

Why real, not synthetic, data. We first built a procedural generator of templated supersession timelines (explicit "X now Y" updates). With the history in context, frontier models score 100% across configurations: they resolve clean, explicit updates by a last-mention scan. Synthetic templated supersession is therefore *saturated* and cannot surface the failure, which is likely why it is under-measured. We accordingly use real conversational data throughout, where updates are implicit and paraphrased. The generator is retained as a control and for the environment's stale-penalty ablation. As Section 5.3 shows, it also serves as a training curriculum whose learned skill transfers back to the real data, turning a saturated probe into a useful teacher.

Data and grading. We use the LongMemEval `knowledge-update` subset. Section 5.1 uses all 78 questions on the *oracle* split (evidence sessions only, $\sim$2 sessions/question), graded by an LLM judge (`gpt-4.1-mini`) following the LongMemEval protocol. Because this judge is itself one of the evaluated models, we cross-check it against the judge-free programmatic matcher (Section 5.2), which agrees to within a few points, guarding against self-grading bias. Section 5.2 uses the same questions on the much longer _*s* split ($\sim$48 sessions, $\sim$122k tokens/question), graded by the programmatic matcher held constant across conditions for a fair comparison. The environment's intrinsic reward uses the same matcher.

Conditions. FULL-CONTEXT places every session in the model context and asks the question (an upper bound where memory is not the bottleneck). BOUNDED-MEMORY is the Supersede agent of Section 3: a notes field of B characters (B=300 unless noted), rewritten one session at a time, with raw sessions never re-fed. For the scale study we vary both the history length (oracle vs. _*s*) and the budget B, including a *constant-ratio* setting in which B grows proportionally with the number of sessions (150 characters per session, matching the oracle ratio).

Models and significance. gpt-4.1-mini and gpt-4.1 at temperature 0, and the frontier reasoning model gpt-5.4 at `reasoning_effort`=low (no temperature). Conditions are run on the same questions, so we report a paired McNemar test (continuity-corrected χ^2).

Training setup. The diagnosis above uses proprietary frontier models, which we cannot fine-tune; to test whether the environment's reward is a usable *learning* signal we therefore switch to an open, trainable model, Qwen2.5-3B-Instruct (Qwen Team, 2025). We train with GRPO (Shao et al., 2024), which estimates its baseline from a group of sampled rollouts rather than a learned value critic (a natural fit for our programmatic, verifiable reward), on the `prime-rl` stack (LoRA, rank 32, $\alpha = 64$ (Hu et al., 2022); learning rate 10^{-5}; batch 32, group 8; 4$\times$ A10G GPUs). Training tasks come from the procedural generator in *bounded-memory* mode (a fixed set of 2000 episodes, disjoint from the real held-out questions) — 6–8 session timelines in which an introduced fact is later superseded and the update is buried among distractor sessions. Two points make this a genuine test rather than a tautology. First, the held-out evaluation is the *real* LongMemEval oracle set (Section 5.1), never seen in training, so we measure transfer from synthetic to real supersession, not memorization. Second, the synthetic saturation noted above is a property of *frontier* models reading the *full* history by last-mention scan; for a small model that must instead *maintain* the fact in a bounded memory, even templated supersession is an unsolved behavior to learn (its untrained reward is far from ceiling, Section 5.3). Base and trained models are graded by the same programmatic matcher used in Section 5.2, so the pre/post comparison is internally exact. All held-out evaluations (base and every checkpoint) use greedy decoding

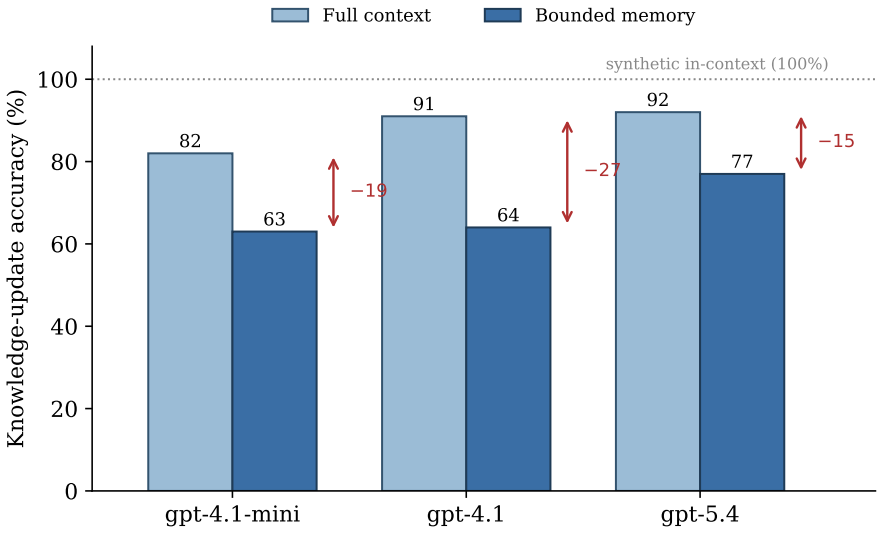

Figure 2: Knowledge-update accuracy ($n = 78$ oracle questions, LongMemEval judge protocol): full context vs. bounded memory, across three models. Stronger models read context better (full-context rises to 92%) but bounded-memory accuracy does not catch up; this supersession gap is the cost of memory maintenance under updates. The dotted line marks synthetic in-context performance (100%), which is saturated and uninformative.

Table 1: LongMemEval `knowledge-update`, oracle split ($n = 78$). The full-context → bounded-memory gap is significant on both the small and the frontier model. (McNemar is not computed for the intermediate gpt-4.1; the small and frontier models bracket the effect.)

Model	Full context	Bounded memory	Paired McNemar
gpt-4.1-mini	82%	63%	$p = 0.0035$
gpt-4.1	91%	64%	—
gpt-5.4	92%	77%	$p = 0.0033$

(temperature 0), so a checkpoint's score is deterministic under re-evaluation: re-running the eval reproduces the same count, and the only stochastic factor in the held-out result is the single training seed, which we flag in Section 7.

5 Results

5.1 Bounded memory degrades supersession

The gap is significant, and a bigger model does not close it. Table 1 and Figure 2 report the result. For gpt-4.1-mini, accuracy falls from 82% to 63%: the paired test counts 19 questions full-context answers correctly but bounded-memory does not, against only 4 in reverse ($p = 0.0035$). On the frontier, gpt-5.4 raises full-context accuracy to 92% but bounded-memory only to 77% ($p = 0.0033$; 13 vs. 1). Across the family, bounded-memory accuracy $(63, 64, 77)$ climbs far more slowly than full-context accuracy $(82, 91, 92)$, which saturates. The bottleneck is memory maintenance, not comprehension, and scaling the model only partially relieves it. (Roughly 13% of questions are wrong even in full context, confirming real updates are genuinely hard.)

Failure modes. Errors are dominated by the relevant fact being compressed away or not overwritten. Asked "How many Korean restaurants have I tried?" (gold: *four*), the agent

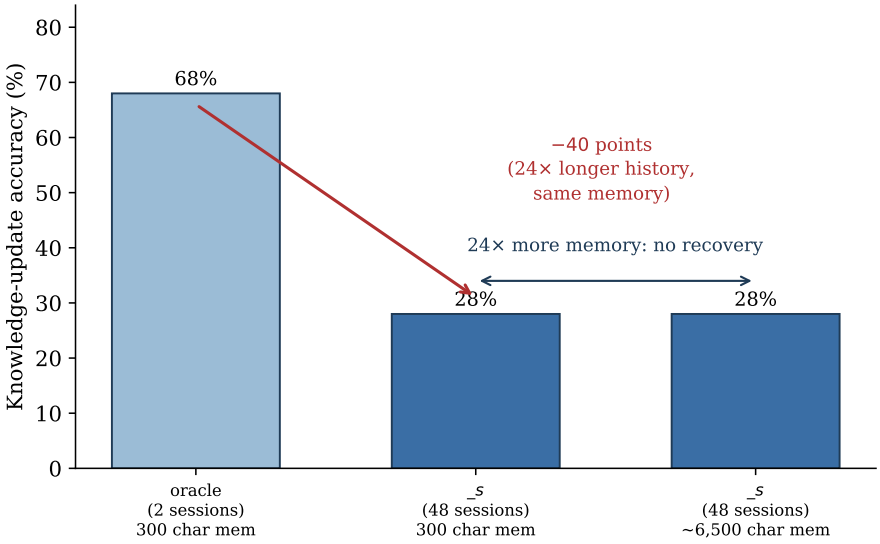

Figure 3: Scale, not size (gpt-4.1-mini, $n = 25$, programmatic matcher held constant across conditions). Growing the conversation 24× at a fixed 300-character memory collapses accuracy (68% → 28%). Giving the agent 24× more memory (constant ratio) recovers none of it (28% → 28%). The failure tracks conversation length, not the compression ratio.

Table 2: Scale study, LongMemEval `knowledge-update` ($n = 25$, gpt-4.1-mini). More history collapses accuracy; proportional memory does not restore it.

Split	History	Memory	Accuracy
oracle	∼2 sessions	300 char	68%
_s	∼48 sessions	300 char	28%
_s	∼48 sessions	∼7,150 char (const. ratio)	28%

answers "you haven't mentioned any." Asked "Where did Rachel move to?" (gold: *the suburbs*), gpt-5.4 answers "no information about Rachel." The model is competent; its memory no longer contains the updated fact.

5.2 It is scale, not memory size

A natural objection to Section 5.1 is that $B = 300$ characters is simply too small, that the failure is an artifact of an undersized memory rather than of supersession. We test this directly by growing the conversation 24× (oracle → _s) and, separately, growing the memory proportionally. Results (gpt-4.1-mini, $n = 25$) are in Table 2 and Figure 3.

More history collapses supersession. At a fixed 300-character memory, lengthening the conversation from ∼2 to ∼48 sessions drops accuracy from 68% to 28% (paired McNemar $p = 0.002$), a 40-point fall driven by relevant facts being squeezed out or overwritten by stale ones across the longer history.

Proportional memory yields no detectable recovery ($n = 25$). Granting the agent 24× more memory (the constant-ratio setting, ∼7,150 characters) leaves accuracy unchanged at 28%. This is not because the budget went unused: every one of the 25 answers differs between the two _s conditions, so the larger memory demonstrably changed the agent's behavior; it simply helped and hurt in equal measure (McNemar $b = c = 4$, no net effect). The failure therefore tracks the *length* of the conversation, not the compression ratio. Combined with Section 5.1, neither a bigger model nor a bigger memory closes the gap.

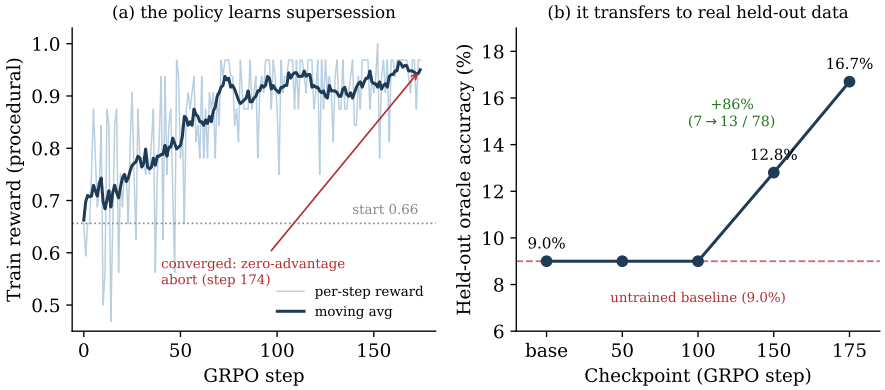

Figure 4: Training closes the gap. **(a)** GRPO reward on procedural episodes rises from 0.66 to 0.97 and then self-terminates by step 175: once nearly every rollout in a group succeeds, the group-relative advantage vanishes and there is no gradient left: the procedural distribution is solved. **(b)** On the *real*, held-out LongMemEval oracle set, the trained policy lifts accuracy from the 9.0% untrained baseline to 16.7%. The rise is monotonic and switches on precisely when the reward saturates (˜step 100): the learned behavior transfers, and is still climbing at the final checkpoint.

Table 3: Held-out gap-closing. Qwen2.5-3B on the LongMemEval `knowledge-update` oracle set ($n = 78$, bounded memory, programmatic matcher), by GRPO checkpoint. Pre and post differ only by the trained LoRA adapter.

Checkpoint	Oracle accuracy	vs. baseline
base (untrained)	$7/78 = 9.0\%$	—
step 50	$7/78 = 9.0\%$	+0.0
step 100	$7/78 = 9.0\%$	+0.0
step 150	$10/78 = 12.8\%$	+3.8
step 175	$\mathbf{13/78 = 16.7\%}$	**+7.7**

Environment validation. Running the released environment end-to-end on all 78 questions, every rollout terminates cleanly and the intrinsic programmatic reward yields 57.7% for gpt-4.1-mini, consistent with the 63% measured by the LLM judge (the matcher is stricter). The environment thus faithfully reproduces the failure it is designed to train against.

5.3 Training closes the gap

Sections 5.1 and 5.2 closed off the two easy escapes: neither a stronger model nor a larger memory removes the gap. The remaining hypothesis is that supersession is a *policy* that must be learned. We test it directly by training on the environment and measuring transfer to the real held-out benchmark.

The policy learns the behavior. On the procedural training distribution, GRPO reward climbs from 0.66 to 0.97 (Figure 4a). The run halts itself at step 175 on ten consecutive zero-advantage batches: once almost every sampled rollout in a group answers correctly, GRPO's group-relative advantage (and hence the gradient) goes to zero. The model has exhausted what the easy episodes can teach. This is a property of the curriculum, not a failure of training, and it upper-bounds what this run can reach (Section 7).

And it transfers to real data. The question is whether anything learned on synthetic timelines survives on real conversations. It does (Table 3, Figure 4b): held-out oracle accuracy rises from 9.0% to 16.7% (13/78 vs. 7/78 correct), a +7.7-point (+86% relative) gain that nearly doubles the untrained baseline (a single run; see Section 7). The *shape* of the

curve is the evidence that this is real learning and not a lucky checkpoint: accuracy is flat through step 100, then rises monotonically (12.8%, 16.7%). It begins to rise exactly when the training reward saturates, i.e. once the behavior is actually acquired. Because the harness, prompt, data split, and matcher are all held fixed, the only thing that changed is the policy; the gain is attributable to training alone. It is still climbing at the last checkpoint before the run self-terminated, so 16.7% is a floor on what this curriculum yields, not a ceiling.

6 Discussion

The results reframe supersession from a comprehension problem to a memory-policy problem. Stronger models read updated context well, but when they must *decide what to keep* under a bounded memory, they discard or fail to overwrite the value that later matters. This is precisely the regime that production memory systems operate in, and precisely where a heuristic UPDATE/DELETE policy (Chhikara et al., 2025) is brittle. Our two scaling axes rule out the easy fixes: the gap does not close as the model grows (Section 5.1), and it does not close as the memory grows (Section 5.2). Waiting for a larger model or simply allocating a larger memory will not solve it, and allocating memory proportional to an ever-growing conversation is itself untenable at scale. What is left is a memory policy *trained* to preserve currency, and Section 5.3 shows this is the axis that moves: the same bounded-memory agent, after GRPO training on Supersede, nearly doubles its held-out accuracy. The lift is partial and the model is small, but the contrast is the point: the two dimensions that *should* have helped (model size, memory size) do not, while the one the environment was built to exercise (a trained currency-preserving policy) does. Supersession is thus best understood not as a capability the next model will absorb, but as a behavior that must be optimized for; the environment and its verifiable reward are what make that optimization possible.

7 Limitations

The training result is a proof of mechanism, not a finished policy. It uses a single small model (Qwen2.5-3B) and a single run; the absolute gain (+7.7 points) is modest and the trained accuracy (16.7%) remains far below the frontier full-context ceiling, as expected for a 3B model and bounded further by the curriculum. Because the procedural episodes are solved before the run ends (training self-terminates on zero advantage, Section 5.3), this curriculum cannot teach harder supersession than it contains; the clear next lever is longer, more implicit training episodes that do not saturate, and we expect them to extend the curve in Figure 4b rather than bend it. What the result does establish is directional and clean: training on the environment moves held-out accuracy monotonically, where model size and memory size did not.

The diagnosis carries its own caveats. The scale study uses $n = 25$ (vs. $n = 78$ for the main gap), so the 40-point drop is robust but the constant-ratio null is measured on a smaller, noisier sample. The additional _s sessions are distractors, so our "scale" axis grows the surrounding conversation rather than the number of updates to the tracked fact; both worsen memory maintenance, but they are not identical. Grading by LLM judge (Section 5.1) introduces some noise (a few failures are judge strictness, e.g. "over 25" vs. gold "25"); the paired significance survives a handful of such flips, and the scale and training studies avoid this by using the programmatic matcher throughout. The two diagnosis scales are themselves graded differently (the frontier gap of Section 5.1 by judge, the trainable result by matcher), so their absolute numbers are not directly comparable; what the claim rests on is the matcher held fixed across the base/trained comparison. That matcher is lenient on surface form (a manual audit of the trained model's matched answers found most carry the gold value verbatim, with a few crediting near-misses); because it grades base and trained identically, the pre/post *delta* is unaffected, though absolute accuracies should be read as matcher-graded. Finally, the training gain is a single run: we offer the monotonic checkpoint curve as evidence that it reflects learning rather than a lucky seed, but its statistical significance is not formally established at $n = 78$, which we address with multi-seed runs in ongoing work.

8 Conclusion

Agents fail at supersession not because they cannot read updates but because they cannot maintain them under a bounded memory. This supersession gap is significant, reproducible on a trusted benchmark, deepens as the conversation grows, and is closed by neither a larger model nor a larger memory. It is, however, closed in part by *training*: a small open model fine-tuned on Supersede nearly doubles its held-out supersession accuracy, monotonically as it learns. We release Supersede (the diagnosis, the open environment, and this first gap-closing result together) as a step toward agents whose memory stays current not by accident of scale, but because they were trained to keep it so.

References

Anthropic. Using Claude's chat search and memory to build on previous context. https://support.anthropic.com/en/articles/11817273, 2026.

Guanzheng Chen, Michael Qizhe Shieh, and Lidong Bing. LongRLVR: Long-context reinforcement learning requires verifiable context rewards. *arXiv preprint arXiv:2603.02146*, 2026. ICLR 2026.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready AI agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*, 2025.

Yiming Du, Baojun Wang, Yifan Xiang, Zhaowei Wang, Wenyu Huang, Boyang Xue, Bin Liang, Xingshan Zeng, Fei Mi, Haoli Bai, Lifeng Shang, Jeff Z. Pan, Yuxin Jiang, and Kam-Fai Wong. Memory-T1: Reinforcement learning for temporal reasoning in multi-session agents. *arXiv preprint arXiv:2512.20092*, 2025.

Google. Personalization and memory in Gemini. https://gemini.google/release-notes/, 2026.

Zexue He, Yu Wang, Churan Zhi, Yuanzhe Hu, Tzu-Ping Chen, Lang Yin, Ze Chen, Tong Arthur Wu, Siru Ouyang, Zihan Wang, Jiaxin Pei, Julian McAuley, Yejin Choi, and Alex Pentland. MemoryArena: Benchmarking agent memory in interdependent multi-session agentic tasks. *arXiv preprint arXiv:2602.16313*, 2026.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022. arXiv:2106.09685.

Yuanzhe Hu, Yu Wang, and Julian McAuley. Evaluating memory in LLM agents via incremental multi-turn interactions. *arXiv preprint arXiv:2507.05257*, 2025.

Hugging Face and Meta PyTorch and contributors. The open source community is backing OpenEnv for agentic RL. https://huggingface.co/blog/openenv-agentic-rl, 2026.

Ruoran Li, Xinghua Zhang, Haiyang Yu, Shitong Duan, Xiang Li, Wenxin Xiang, Chonghua Liao, Xudong Guo, Yongbin Li, and Jinli Suo. MemPO: Self-memory policy optimization for long-horizon agents, 2026.

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. arXiv:2402.17753.

OpenAI. Dreaming: Better memory for a more helpful ChatGPT. https://openai.com/index/chatgpt-memory-dreaming/, 2026. Self-reported internal evaluation; methodology and dataset not released.

Atahan Özer and Çağatay Yıldız. Question answering under temporal conflict: Evaluating and organizing evolving knowledge with LLMs. *arXiv preprint arXiv:2506.07270*, 2025.

Prime Intellect. Environments hub: A community hub to scale RL to open AGI. https://www.primeintellect.ai/blog/environments, 2025. Open registry and library (verifiers) for reinforcement-learning environments.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. Introduces Group Relative Policy Optimization (GRPO).

Haoran Tan, Zeyu Zhang, Chen Ma, Xu Chen, Quanyu Dai, and Zhenhua Dong. MemBench: Towards more comprehensive evaluation on the memory of LLM-based agents. In *Findings of the Association for Computational Linguistics: ACL 2025*, 2025. arXiv:2506.21605.

Md Nayem Uddin, Kumar Shubham, Eduardo Blanco, Chitta Baral, and Gengyu Wang. From recall to forgetting: Benchmarking long-term memory for personalized agents. *arXiv preprint arXiv:2604.20006*, 2026. Introduces the Memora benchmark and the Forgetting-Aware Memory Accuracy (FAMA) metric.

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. arXiv:2410.10813.

Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, and Hao Zhou. MemAgent: Reshaping long-context LLM with multi-conv RL-based memory agent. *arXiv preprint arXiv:2507.02259*, 2025.

Yi Yu, Liuyi Yao, Yuexiang Xie, Qingquan Tan, Jiaqi Feng, Yaliang Li, and Libing Wu. Agentic memory: Learning unified long-term and short-term memory management for large language model agents. *arXiv preprint arXiv:2601.01885*, 2026.

A Reward matcher

The environment scores answers with a programmatic, judge-free matcher, so it can be trained and evaluated without an auxiliary model. Given a model answer a and a gold (current) value g:

1. **Normalize.** Both strings are lowercased, punctuation is replaced by spaces, and runs of whitespace are collapsed.
2. **Gold variants.** g is split on connectives ("or", "and", "/", ";", ",") and each normalized piece of length ≥ 2 is kept as an acceptable variant (so "25 minutes and 50 seconds" also accepts "25 minutes" or "50 seconds").
3. **Match.** The answer matches if any gold variant occurs as a normalized substring of a.
4. **Fallback.** If no variant matches, a token-overlap fallback fires: the answer matches when its token overlap with a gold variant exceeds 0.6.

$r_{\text{cur}} = \mathbb{1}[\text{match}]$. When the superseded values are known, the same matcher is run against each stale value; a positive match there triggers the penalty r_{stale}. The matcher is intentionally lenient on surface form but strict on *which* value is reported: the quantity supersession turns on.

B Training details

Table 4 lists the full GRPO configuration. Training uses procedurally generated bounded-memory episodes (6–8 sessions, one introduced fact later superseded, the update buried

among distractor sessions); evaluation is the untouched real LongMemEval oracle set. The run self-terminates at step 174 after ten consecutive zero-advantage batches: every sampled rollout in the group succeeds, so GRPO's group-relative advantage (and the gradient) vanishes.

Table 4: GRPO training configuration.

Setting	Value
Base model	Qwen2.5-3B-Instruct
Algorithm	GRPO (Shao et al., 2024)
Adapter	LoRA (Hu et al., 2022), rank 32, $\alpha = 64$, dropout 0
Target modules	{q,k,v,o,gate,up,down}_proj
Bias	none
Learning rate	10^{-5}
Batch size / group size	32 / 8
Hardware	4× A10G
Stack	prime-rl / verifiers
Training episodes	procedural, 6–8 sessions (2000 total)
KL penalty	none ($\beta = 0$)
Decoding (eval)	greedy (temperature 0); answer $\leq$ 64 tokens
Termination	self-halt at step 175 of 200 max (10 zero-adv. batches)

C Example episodes

Two procedurally generated training episodes. The agent sees one session at a time and maintains a bounded notes memory; raw sessions are never re-fed. The reward is 1 only if the final answer conveys the *current* value (in bold), not a superseded one.

Example 1 (single update, 7 sessions).

- S1: "I recently settled in Chicago."
- S2–S3: distractors (trip planning; a new book)
- S4: "I moved again; I'm in Atlanta now."
- S5–S7: distractors (desk; weather; a recipe)
- **Query:** "Which city do I currently live in?" — current: **Atlanta**; superseded: Chicago.

Example 2 (two updates, 6 sessions).

- S1: "I got myself a Mazda."
- S2–S3: distractors (trip planning; a new hobby)
- S4: "New ride update: it's a Kia now."
- S5: "I switched to a Lexus now."
- S6: distractor (a new book)
- **Query:** "What car do I currently drive?" — current: **Lexus**; superseded: Mazda, Kia.