MIRAGE: Memory Revision and Path-Aware Retrieval for Conflict Resolution in LLM Agent Memory

Anonymous submission

Abstract

LLM agent memory systems increasingly rely on non-parametric state to reuse experience across tasks. However, memory maintenance remains under-specified: append-only storage can leave stale evidence, conflicting facts, and disconnected retrieval snippets in the active context. We study memory-state maintenance as an alternative to append-only context construction: deciding what evidence should remain active, what should be suppressed, and what structure is needed for multi-hop reuse when memory updates follow an explicit authority signal. We propose MIRAGE, a memory-maintenance framework that maintains agent memory through ordered revision, provenance marking, and path-aware retrieval. In this paper, we evaluate three concrete MIRAGE paths on MemoryAgentBench FactConsolidation: ordered conflict revision (MIRAGE-R), provenance-marked revision (MIRAGE-P), and graph/path-aware retrieval (MIRAGE-G). On the 6k multi-hop setting, MIRAGE-R improves accuracy on 100 questions from 0.28 to 0.68 while reducing token use by 32.8%; on the 32k multi-hop setting, accuracy improves from 0.21 to 0.65 with a 34.5% token reduction. On the same 6k multi-hop setting, MIRAGE-P maintains comparable accuracy at 0.71 while reducing empty outputs, and MIRAGE-G raises retrieval-based accuracy from 0.17 for lexical top-k retrieval to 0.59 while preserving an 80.1% token reduction. Our findings suggest that MIRAGE provides a practical memory-maintenance layer for long-context agents, especially in settings where persistent state must be revised, audited, and retrieved through multi-hop evidence rather than simply accumulated.

Introduction

Large language model (LLM) agents are increasingly expected to operate beyond one-shot problem solving. In long-running settings, they accumulate interaction traces, user feedback, tool-use outcomes, and successful or failed trajectories as reusable non-parametric state. This shift appears across reasoning-and-acting agents (Yao et al. 2022), generative agents (Park et al. 2023), reflective agents (Shinn et al. 2024), experiential-learning agents (Zhao et al. 2024), workflow memory (Wang et al. 2024), long-term memory systems (Zhong et al. 2024), and broader visions of self-evolving agents (Gao et al. 2025). As a result, memory is no longer a passive log of past interactions; it becomes an active substrate that determines what evidence the agent can reuse, revise, suppress, and expose at inference time.

Persistence also turns memory into a maintenance problem. If memory is append-only, old and new evidence remain active together. The model must then search a long context and resolve conflicts during inference. Because long-context models use evidence unevenly across input positions (Liu et al. 2023), leaving stale evidence in the active context increases the burden of conflict resolution at inference time rather than resolving conflicts through a memory-state update. A natural response to this context burden is to retrieve fewer snippets. Retrieval-augmented generation shows that non-parametric evidence can be selectively supplied to the model (Lewis et al. 2020). However, multi-hop question answering requires connected supporting facts, as formalized by HotpotQA (Yang et al. 2018), and IRCoT shows that retrieval for multi-step questions can benefit from being interleaved with reasoning (Trivedi et al. 2023). Thus lexical top-k retrieval may save tokens while breaking the evidence chains needed for multi-hop memory reuse.

We therefore study memory maintenance for LLM agents. The question is: *Given a pool of potentially conflicting evidence, which evidence should remain active, which evidence should be suppressed, and what structure should retrieval preserve?* Our approach is to treat memory as a maintained state rather than an append-only context: conflicts are resolved through explicit revision, superseded evidence is suppressed with provenance, and retrieval preserves connected evidence paths for multi-hop reuse. We focus on these memory-maintenance components rather than broader claims about fully self-evolving agents.

We propose MIRAGE, a memory-maintenance framework that separates revision, provenance, and retrieval through three complementary paths: **MIRAGE-R** resolves ordered conflicts by keeping the newest authoritative evidence, **MIRAGE-P** records superseded evidence for auditability, and **MIRAGE-G** retrieves connected relation paths instead of isolated high-overlap snippets. We then evaluate MIRAGE on MemoryAgentBench FactConsolidation (Hu, Wang, and McAuley 2025), where explicit serial order defines authoritative facts and the raw-context baseline receives the same or-

This is an anonymized submission for review purposes only. Distribution, citation, or public sharing of this manuscript is strictly prohibited. Copyright and publication details will appear in the final version if accepted.

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.

dering rule. The results show that ordered revision improves multi-hop accuracy while substantially reducing token usage, provenance-marked revision preserves auditability with comparable answer quality, and path-aware retrieval recovers much of the accuracy lost by lexical top-k retrieval while preserving large context savings.

Overall, our contributions are summarized as follows:

- **Memory-maintenance formulation.** We formulate persistent agent memory as a maintained state rather than an append-only context, identifying stale-evidence suppression and evidence-path preservation as two core operations for reliable multi-hop reuse.
- **The MIRAGE framework.** We introduce MIRAGE, a memory-maintenance framework that combines ordered conflict revision, provenance-marked suppression, and graph/path-aware retrieval. These components respectively decide which evidence remains active, keep revisions auditable, and preserve connected evidence paths during retrieval.
- **Empirical evidence on FactConsolidation.** We evaluate MIRAGE on MemoryAgentBench FactConsolidation under a controlled authority-order setting. The results show that ordered revision improves multi-hop accuracy while substantially reducing token usage, provenance marking maintains comparable answer quality with fewer empty outputs, and path-aware retrieval recovers much of the accuracy lost by lexical top-k retrieval while preserving large context savings.

Related Work

Experience-driven LLM agents. A growing line of LLM-agent work studies how agents reuse experience without updating model parameters. Reflexion stores verbal reflections from task feedback for later trials (Shinn et al. 2024). Generative Agents maintain a memory stream and retrieve observations for planning and reflection (Park et al. 2023). ExpeL extracts reusable insights from successful and failed trajectories (Zhao et al. 2024). Agent Workflow Memory induces reusable workflows from past web-agent trajectories (Wang et al. 2024). These works show that traces and feedback can guide future behavior, but their main focus is experience acquisition and reuse. MIRAGE instead asks how accumulated experience should be maintained when older evidence is superseded by newer evidence.

Memory consolidation and maintenance. Long-term agent memory systems have begun to study how memory should be stored, updated, forgotten, and compacted. MemoryBank introduces explicit long-term memory with updating and forgetting mechanisms for sustained interaction (Zhong et al. 2024). MemGPT applies operating-system-style memory management to move information between context and external storage (Packer et al. 2023). Recent consolidation methods further study recurrence-triggered memory construction, residual experience organization, and offline consolidation (Dai et al. 2026; Tan et al. 2026; Ye et al. 2026). Benchmarks and reliability studies evaluate memory update quality, forgetting, and long-horizon memory behavior (Hu, Wang, and McAuley 2025; Yang et al. 2026; Uddin et al. 2026; Zhao et al. 2026). MIRAGE is closest to this maintenance-oriented line, but narrows the focus to active-state revision before inference: under an explicit authority order, stale evidence should be suppressed, retained evidence should remain auditable, and unresolved superseded alternatives should be kept out of the active context.

Path-aware retrieval for multi-hop reuse. Retrieval-augmented generation established non-parametric retrieval as a way to provide external evidence to language models (Lewis et al. 2020). However, multi-hop QA benchmarks such as HotpotQA require connected supporting facts rather than individually relevant but disconnected snippets (Yang et al. 2018). Multi-hop retrieval systems address this by recursively retrieving evidence across hops (Xiong et al. 2020), interleaving retrieval with reasoning, as in IRCoT (Trivedi et al. 2023), or by condensing retrieved evidence across hops, as in Baleen (Khattab, Potts, and Zaharia 2021). Graph-based retrieval systems such as HippoRAG show that structured knowledge graphs can improve multi-hop evidence retrieval (Gutiérrez et al. 2024). MIRAGE-G follows the same path-preservation intuition, but applies it to revised agent memory rather than a static external corpus: after stale facts are removed from active state, retrieval must still expose the relation chains needed for multi-hop reuse.

Problem Formulation

We study an LLM agent with frozen model parameters and mutable non-parametric memory. For each benchmark instance, the agent receives a fact pool $F = \{f_i\}_{i=1}^{n}$ and a question q. Each evidence item is represented as

$$f_i = (\tau_i, x_i, \kappa_i), \tag{1}$$

where x_i is the fact text, τ_i is an externally supplied authority order, and $\kappa_i = \kappa(x_i)$ is a conflict key. The key denotes the mutable slot described by the fact, such as a subject–relation pair. For example, two facts about the capital of the same country share the same key even if they give different values. In FactConsolidation, τ_i is the serial number attached to the fact, and a larger serial number supersedes a smaller one for the same key.

An append-only memory exposes all facts to the model:

$$C_{\mathrm{raw}}(q) = F. \tag{2}$$

This leaves conflict resolution to inference: the model must identify all facts with the relevant key, compare their authority orders, and still follow any multi-hop chain needed to answer q. Consequently, stale facts remain in the active context, competing with the current fact and thereby contaminating the context. This motivates treating conflict resolution as a memory-maintenance operation before inference, rather than as an additional reasoning burden inside the prompt.

The problem studied in this paper is whether these operations can be handled before inference as memory maintenance: selecting current evidence, suppressing superseded evidence, preserving revision provenance, and retrieving connected evidence paths required for multi-hop reasoning. MIRAGE instantiates these operations through ordered revision, provenance marking, and path-aware retrieval.

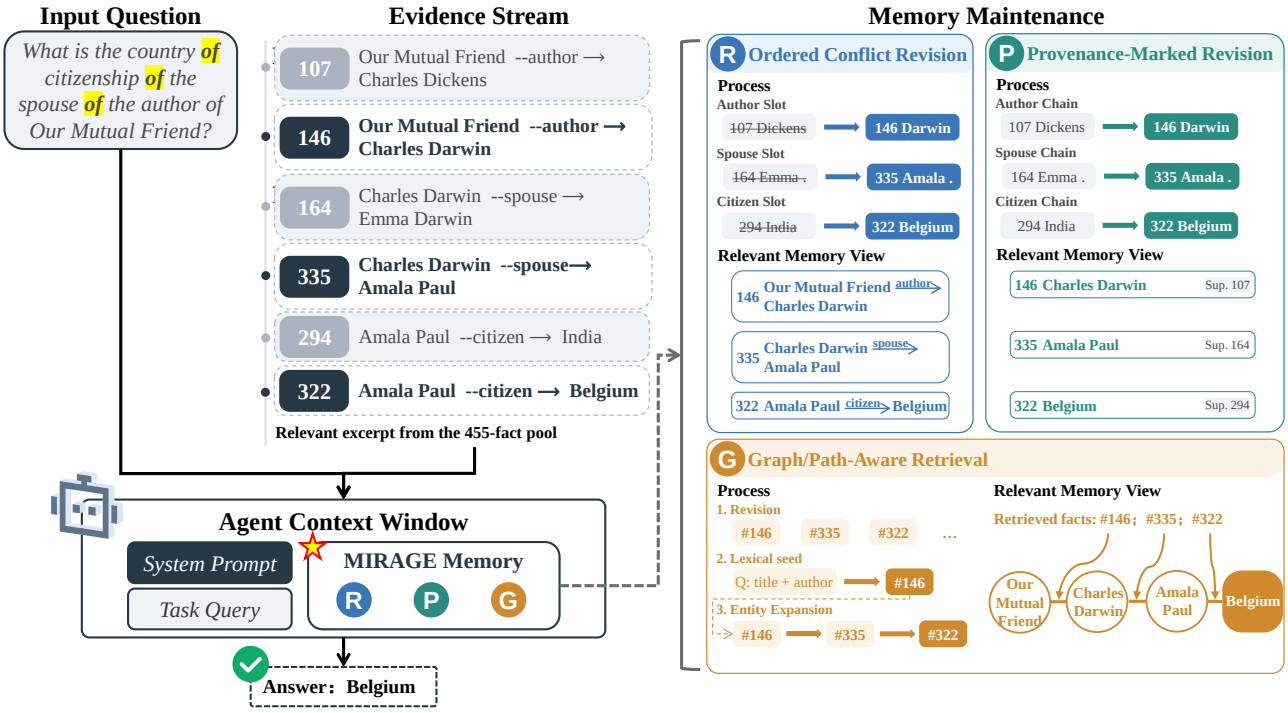


Figure 1: Overview framework of MIRAGE. Given an input question (multi-hop) and the evidence stream, MIRAGE maintains agent memory through three mechanisms: ordered conflict revision selects the current authoritative facts, provenance-marked revision records superseded evidence for auditability, and graph/path-aware retrieval preserves multi-hop evidence chains before injecting the memory into the agent context.

Method

MIRAGE converts an append-only fact pool into a maintained memory state before the model is queried. As illustrated in Figure 1, it starts from the same input question and evidence stream, then exposes three optional maintenance views for the agent context. **Revision** decides which evidence remains active under the authority order. **Provenance** records what the active evidence supersedes without reintroducing the full stale context. **Path-aware retrieval** selects a compact subset of active evidence while preserving relation chains needed for multi-hop reuse. These operations correspond to MIRAGE-R, MIRAGE-P, and MIRAGE-G.

MIRAGE-R: Ordered Conflict Revision

MIRAGE-R turns conflict resolution from an in-context reasoning burden into a state update. Given facts $F = \{f_i\}_{i=1}^{n}$ and a key function $\kappa(\cdot)$, it groups facts by conflict key and keeps the highest-authority fact in each group. Let $K = \{\kappa_i \mid f_i \in F\}$ be the set of conflict keys. MIRAGE-R constructs a key-indexed active state:

$$S(k) = \arg \max_{f_i : \kappa_i = k} \tau_i, \quad k \in K. \tag{3}$$

The rendered active memory contains the retained facts

$$A = \{S(k) \mid k \in K\}, \tag{4}$$

and the suppressed set is

$$D = \{f_i \mid \exists f_j : \kappa_i = \kappa_j \wedge \tau_j > \tau_i\}. \tag{5}$$

In the evaluated FactConsolidation setting, τ_i is the fact serial number, so Algorithm 1 can implement revision in one pass over the numbered facts. Conflict keys are obtained with a deterministic template parser over the FactConsolidation fact format; the parser uses only the fact text and serial number, not the question or answer. The raw-context baseline receives the same instruction that larger serial numbers override smaller ones. Therefore, the comparison tests whether applying the authority rule as a memory-state update is more reliable than leaving the same rule to inference.

MIRAGE-P: Provenance-Marked Revision

Revision improves compactness, but current-only rendering hides why a fact is current and what it replaced. MIRAGE-P keeps the same active facts as MIRAGE-R and adds compact provenance markers. For each conflict key k, we define the provenance set of the active fact as

$$P(k) = \{f_i \mid \kappa_i = k \wedge \tau_i < \tau(S(k))\}. \tag{6}$$

MIRAGE-P renders $S(k)$ with the serial numbers of facts in $P(k)$, for example `[CURRENT; SUPERSEDES 17, 23]`. Facts with no superseded alternatives are marked as current without a supersession list. This representation exposes revision history for audit and possible correction while

Algorithm 1: MIRAGE-R Ordered Conflict Revision

Input: numbered facts $F = \{(\tau_i, x_i)\}$, key function $\kappa(\cdot)$
$S \leftarrow \emptyset$; $D \leftarrow \emptyset$
for each fact $(\tau_i, x_i) \in F$ in serial order **do**
  $k \leftarrow \kappa(x_i)$
  if $k \notin S$ or $\tau_i > \tau(S[k])$ **then**
    if $k \in S$ **then**
      $D \leftarrow D \cup \{S[k]\}$
    end if
    $S[k] \leftarrow (\tau_i, x_i)$
  else
    $D \leftarrow D \cup \{(\tau_i, x_i)\}$
  end if
end for
return active facts S and suppressed facts D

avoiding the main failure mode of append-only memory: the stale facts themselves are not placed back into the active context as competing evidence.

MIRAGE-G: Path-Aware Retrieval

MIRAGE-G addresses the complementary problem that compact retrieval can break multi-hop evidence chains. It operates after ordered revision, so retrieval is performed over the active facts rather than the raw conflicting pool. When a retained fact can be parsed as a subject–relation–object triple (u_i, r_i, v_i), MIRAGE-G adds it as an edge in a consolidated relation graph:

$$G_A = (V_A, E_A), \quad E_A = \{(u_i, r_i, v_i, f_i) \mid f_i \in A\}. \quad (7)$$

For a question q, lexical retrieval first selects seed facts by overlap with question terms. MIRAGE-G then expands from the subject and object entities of these seeds through neighboring subject/object links in G_A using bounded breadth-first search. The output is a retrieved active subset

$$R_{\text{path}}(q) \subseteq A, \quad (8)$$

rendered in fact order. In the reported 100-question run, we use 16 lexical seed facts, expand up to 4 hops, and cap the retrieved context at 96 facts. This design keeps the large token savings of retrieval while making it less likely that a necessary intermediate fact is dropped. The current graph parser is template-based and covers the relation templates used by FactConsolidation; it is not a learned open-domain relation extractor.

Design Rationale and Scope

MIRAGE deliberately separates revision, provenance, and retrieval because they optimize different trade-offs. MIRAGE-R is compact and accurate when the authority order is reliable, but it hides the revision chain. MIRAGE-P spends more tokens to make the retained state auditable. MIRAGE-G is much cheaper than rendering the full revised state, but it can fail if relation parsing misses an edge or if the hop and fact budgets prune a required path. Reporting these components separately exposes the accuracy–cost–auditability frontier instead of presenting MIRAGE as a single monolithic prompt transformation.

The current implementation is intentionally scoped to controlled memory maintenance. It does not infer authority: the benchmark already defines serial order, and MIRAGE uses that order as an input. It does not learn free-form relation extraction: the graph is built with template rules that match the benchmark format. These restrictions isolate whether stale-evidence suppression and path-preserving retrieval help under controlled conditions, before evaluating them in more realistic settings where authority and relation structure must be inferred.

Complexity and Context Cost

Once conflict keys are available, ordered revision is linear in the number of facts: each fact updates one entry in the active map S and may add one item to the suppressed set D. Provenance marking adds a small amount of rendered text per revised key. Path-aware retrieval is bounded by the seed count, hop depth, and fact cap; in the reported configuration, at most 96 facts are injected for a question regardless of the raw fact-pool size. These bounds matter because MIRAGE aims to expose a smaller and more coherent inference context, not to build a larger memory prompt. The dominant runtime cost in our experiments remains LLM inference, so we report token usage rather than CPU time.

Experiments

Experimental Setup

Benchmarks and Metrics We evaluate on the FactConsolidation subset of MemoryAgentBench (Hu, Wang, and McAuley 2025), which is designed for Conflict Resolution (CR). The benchmark is constructed from counterfactual edit pairs in MQUAKE (Zhong et al. 2023): each pair contains a true fact and a rewritten, contradictory version, with the rewritten version ordered after the original to simulate factual updates. In the benchmark prompt, each fact is given a serial number, and newer facts have larger serial numbers. Agents are instructed to prioritize later information in case of conflict and answer based on the final memory state. This setting is well suited to our study because it provides an explicit authority signal, allowing us to isolate memory maintenance from learned authority inference.

We use single-hop 6k and multi-hop 6k/32k/64k settings. The 6k, 32k, and 64k labels denote the approximate raw fact-pool context scale used in our runs. All reported main and component runs use 100 questions. Following MemoryAgentBench, we report accuracy using substring exact match for the fact_sh and fact_mh tasks. We also report empty-output count, total token usage, and token reduction relative to raw context. We count an output as empty when the same answer extractor returns no answer text.

Baselines and Compared Variants We compare raw append-only memory with full-state MIRAGE variants and retrieval variants over the consolidated current facts. **Raw context** renders the original conflicting fact list and uses the benchmark instruction that newer facts have larger serial numbers and should be prioritized in case of conflict. **MIRAGE-R** applies ordered conflict revision and renders

Task	Scale	Method	Acc.↑	Empty↓	ΔTok. ↓
Single-hop	6k	Raw context	1.00	0	–
		MIRAGE-R	0.98	2	−33.9%
		MIRAGE-P	0.97	2	−8.6%
		Lex. top-k	0.92	7	−88.4%
		MIRAGE-G	0.93	7	−85.9%
Multi-hop	6k	Raw context	0.28	71	–
		MIRAGE-R	0.68	31	−32.8%
		MIRAGE-P	0.71	25	−8.0%
		Lex. top-k	0.17	70	−81.5%
		MIRAGE-G	0.59	39	−80.1%
	32k	Raw context	0.21	75	–
		MIRAGE-R	0.65	29	−34.5%
		MIRAGE-P	0.64	30	−7.7%
		Lex. top-k	0.02	90	−96.6%
		MIRAGE-G	0.39	54	−93.6%
	64k	Raw context	0.17	76	–
		MIRAGE-R	0.64	34	−36.3%
		MIRAGE-P	0.64	31	−7.6%
		Lex. top-k	0.04	82	−98.0%
		MIRAGE-G	0.27	68	−96.6%

Table 1: Main FactConsolidation results. Acc. is substring exact-match accuracy, Empty counts cases where the shared answer extractor returns no answer text, Lex. top-k denotes lexical top-k retrieval, and ΔTok. is the percentage change in total token usage relative to Raw context within each setting; negative values indicate token reductions.

only the current fact for each conflict key. **MIRAGE-P** renders the same current facts as MIRAGE-R but adds compact supersession markers. **Lexical top-k** first applies MIRAGE-R and then retrieves the top-k current facts by lexical overlap with the question; we set $k = 12$ for all lexical retrieval runs. **MIRAGE-G** also operates over the consolidated current facts, but retrieves path-expanded evidence over the consolidated fact graph rather than isolated high-overlap facts.

Implementation Details All conditions use the same base model, Qwen3.6-27B, with temperature 0.0 and a 1024-token completion budget. The model context window covers the largest 64k raw-context setting used in our experiments. The prompt template and decoding settings are fixed across conditions; only the rendered knowledge pool changes.

All variants are evaluated on the same 100 questions in each setting. Token usage denotes total model token usage accumulated over the 100-question run, and token reduction is computed relative to the corresponding raw-context condition. Empty outputs are counted when the shared answer extractor returns no answer text after using the same extraction procedure across methods. For retrieval variants, lexical retrieval and graph expansion are performed before the LLM call and do not use the gold answer.

Main Results

Table 1 summarizes the main FactConsolidation results. We highlight four findings as follows.

Single-hop sanity check. The single-hop setting serves as a sanity check: raw context already reaches perfect accuracy, and MIRAGE-R and MIRAGE-P preserve near-perfect accuracy while reducing token use by 33.9% and 8.6%, respectively. Compact retrieval also remains viable in this easier setting, with Lexical top-k and MIRAGE-G reaching 0.92 and 0.93 accuracy while reducing token use by more than 85%. This indicates that the main challenge is not single-hop lookup, but conflict resolution and evidence-chain preservation in multi-hop settings.

Ordered revision vs. raw context. The multi-hop settings expose the core memory-maintenance failure. Raw context receives the same serial-order instruction as MIRAGE-R, yet its accuracy drops from 0.28 at 6k to 0.17 at 64k, with 71–76 empty outputs out of 100 questions. This suggests that the model often fails to resolve conflicts and follow evidence chains inside a long append-only prompt, even when the needed facts and update rule are present. MIRAGE-R consistently converts this inference-time burden into a state update: it improves accuracy by 40–47 percentage points across 6k, 32k, and 64k while reducing token use by 32.8–36.3%.

Auditability with provenance. MIRAGE-P adds audit-oriented provenance to the revised state and tests whether this additional revision history preserves answer quality. Across all multi-hop scales, provenance marking keeps accuracy close to MIRAGE-R: 0.71 versus 0.68 at 6k, 0.64 versus 0.65 at 32k, and 0.64 versus 0.64 at 64k. It also keeps empty-output counts close to MIRAGE-R, with fewer empty outputs at 6k and 64k. The trade-off is lower compression: because supersession markers retain revision history, token reduction is only 7.6–8.0% on multi-hop settings. We therefore interpret MIRAGE-P as an auditability-oriented variant rather than the most compact representation.

Path-aware retrieval vs. lexical top-k. The retrieval results show that token reduction alone is not sufficient. Lexical top-k retrieval is extremely compact, but it collapses on multi-hop questions: accuracy falls to 0.17, 0.02, and 0.04 at 6k, 32k, and 64k. MIRAGE-G uses a similarly compact token budget while preserving relation paths, improving accuracy to 0.59, 0.39, and 0.27 on the same settings; Table 2 further shows that the gain remains under the same 12-fact cap. MIRAGE-G remains below full revised-state rendering, but this is expected because it exposes only a compact retrieved subset rather than the full active state. Its purpose is not to replace MIRAGE-R or MIRAGE-P, but to test whether compact retrieval can preserve the evidence structure needed for multi-hop reuse. The consistent gains over Lexical top-k support the retrieval claim: compact agent memory should preserve connected evidence paths, not merely high-overlap snippets.

Figure 2 visualizes the same trade-off from Table 1. MIRAGE-R is the more compact full-state representation with consistently high accuracy, while MIRAGE-P spends more tokens to preserve revision provenance and achieves comparable or slightly higher accuracy in some settings. Lexical top-k moves far right but falls to low accuracy, whereas MIRAGE-G stays near the same compression regime and re-

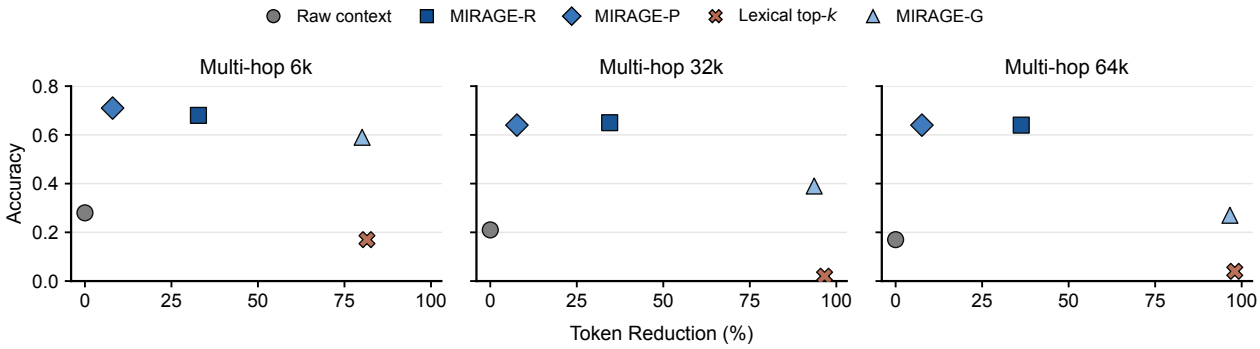


Figure 2: Accuracy–token trade-off on multi-hop FactConsolidation. MIRAGE-R and MIRAGE-P occupy the high-accuracy region, with MIRAGE-P spending more tokens to preserve revision provenance. Lexical top-k retrieval achieves the largest compression but loses multi-hop evidence chains. MIRAGE-G recovers substantial accuracy over lexical retrieval while retaining a compact context.

Method	Seeds	Hops	Cap	Acc.↑	Empty↓	ΔTok.↓
Lex. top-k	–	–	12	0.17	70	−81.5%
Path retrieval	8	3	12	0.41	50	−83.8%
	4	3	48	0.51	45	−83.8%
	8	3	48	0.51	46	−82.1%
	12	3	48	0.54	43	−81.5%
	16	2	96	0.54	43	−80.5%
	16	4	96	0.59	39	−80.1%

Table 2: Path-retrieval ablation on the multi-hop 6k FactConsolidation setting. Seeds is the number of lexical seed facts, Hops is the maximum graph-expansion depth, and Cap is the maximum number of retrieved facts.

covers a substantial fraction of the lost accuracy. This frontier view clarifies why MIRAGE separates revision, provenance, and retrieval rather than reporting a single memory representation.

Path-Retrieval Ablation

The main results show that MIRAGE-G improves over lexical top-k retrieval under compact token budgets. We next ablate the retrieval parameters to test whether this gain depends on a particular configuration. On the multi-hop 6k FactConsolidation setting, we vary the number of lexical seed facts, the maximum graph-expansion depth, and the cap on retrieved facts.

Table 2 verifies that MIRAGE-G is robust across retrieval budgets. With the same 12-fact cap as Lexical top-k, path-aware retrieval improves accuracy from 0.17 to 0.41 and reduces empty outputs from 70 to 50. Increasing the cap further improves accuracy to 0.51–0.59 while keeping token reduction close to Lexical top-k; all path-aware variants reduce token use by 80.1–83.8%, compared with 81.5% for Lexical top-k.

Analysis

Conflict resolution should be a memory operation, not only an inference instruction. The raw-context baseline is not uninformed: it receives the same serial-order rule as MIRAGE-R and contains the facts needed to answer the questions. Its failure therefore cannot be attributed to missing evidence or the absence of the update rule. The more plausible explanation is that append-only memory forces the model to search a long fact pool, resolve conflicts among superseded and current facts, and follow multi-hop evidence chains at the same time. MIRAGE-R improves performance because it moves the conflict-resolution step out of the prompt and into the memory state. In this sense, ordered revision is not merely a compression heuristic; it changes where the update rule is applied.

Suppression is different from forgetting. MIRAGE does not equate suppressing evidence with deleting it. Superseded facts are removed from the active inference context, but their revision history can still be recorded for audit, debugging, or later correction. This distinction matters for persistent agents: a memory can be inactive for answering while still being useful for inspecting why the current state changed. MIRAGE-R tests the value of a clean active state, whereas MIRAGE-P tests whether revision history can be exposed without reintroducing stale facts as competing evidence. The results suggest that provenance can preserve comparable answer quality, but it should be understood as an auditability mechanism rather than a stronger compression mechanism.

Path-aware retrieval is structured compression. The lexical top-k baseline shows that shortening the context is not sufficient. It removes many tokens, but it can also remove bridge facts that do not share surface words with the question. This is especially damaging in multi-hop settings, where the answer depends on a chain of retained facts rather than a single highly overlapping snippet. MIRAGE-G improves over lexical retrieval because it expands from lexical seeds through entity links in the consolidated graph; the same-cap ablation supports that this gain is not just a larger retrieval

budget. The remaining gap between MIRAGE-G and full revised-state rendering is also informative: graph retrieval can miss evidence when relation parsing is incomplete or when the hop and fact budgets prune a required path. Thus MIRAGE-G should be read as a compact retrieval mode, not as a replacement for the full revised state in accuracy-critical settings.

Choosing among MIRAGE-R, MIRAGE-P, and MIRAGE-G. The three MIRAGE paths expose different points on the accuracy–token–auditability frontier. MIRAGE-R represents the compact full-state option: it applies the authority rule before inference and removes superseded competitors from the active context. MIRAGE-P represents the provenance-oriented option: it keeps the revised state inspectable at the cost of weaker compression. MIRAGE-G represents the high-compression option: it is useful when the context budget is tight and the task is compositional, but it remains less accurate than full revised-state rendering. The practical implication is not that one representation dominates, but that reliable agent memory should expose active/suppressed state, provenance, and path-preserving retrieval as separate controls rather than collapsing them into one append-only memory prompt.

Limitations

Our evaluation assumes an explicit authority signal. FactConsolidation provides serial order, and MIRAGE uses this order directly rather than inferring authority from source reliability, timestamps, user corrections, or tool feedback. This isolates the value of memory-state maintenance, but it also means that MIRAGE does not yet solve the harder problem of authority estimation in open-ended agent environments.

Finally, our experiments use one benchmark family and one base model. The results therefore show a controlled proof of concept rather than a universal claim about all long-term agent memory systems. Broader evaluation should include interactive tool agents, personalized memory, noisy user feedback, and multiple model families.

Conclusion

We study memory maintenance for LLM agents under explicit authority order. MIRAGE treats persistent memory as a maintained state rather than an append-only context, using ordered revision to suppress stale evidence, provenance marking to keep revisions auditable, and path-aware retrieval to preserve multi-hop evidence chains. On MemoryAgentBench FactConsolidation, MIRAGE-R consistently improves multi-hop accuracy while reducing token use, MIRAGE-P preserves comparable answerability with auditability, and MIRAGE-G recovers much of the accuracy lost by lexical retrieval under compact context budgets. These results suggest that reliable agent memory requires revision and structure, not only accumulation.

References

Dai, Z.; Deng, S.; Guan, S.; Tian, Y.; Yao, X.; Yan, X.; and Cheng, J. 2026. RecMem: Recurrence-based Memory Consolidation for Efficient and Effective Long-Running LLM Agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, 32353–32376.

Gao, H.-a.; Geng, J.; Hua, W.; Hu, M.; Juan, X.; Liu, H.; Liu, S.; Qiu, J.; Qi, X.; Wu, Y.; et al. 2025. A Survey of Self-Evolving Agents: What, When, How, and Where to Evolve on the Path to Artificial Super Intelligence. *arXiv preprint arXiv:2507.21046*.

Gutiérrez, B. J.; Shu, Y.; Gu, Y.; Yasunaga, M.; and Su, Y. 2024. Hipporag: Neurobiologically inspired long-term memory for large language models. *Advances in neural information processing systems*, 37: 59532–59569.

Hu, Y.; Wang, Y.; and McAuley, J. 2025. Evaluating memory in llm agents via incremental multi-turn interactions. *arXiv preprint arXiv:2507.05257*.

Khattab, O.; Potts, C.; and Zaharia, M. 2021. Baleen: Robust multi-hop reasoning at scale via condensed retrieval. *Advances in Neural Information Processing Systems*, 34: 27670–27682.

Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.-t.; Rocktäschel, T.; et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474.

Liu, N. F.; Lin, K.; Hewitt, J.; Paranjape, A.; Bevilacqua, M.; Petroni, F.; and Liang, P. 2023. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*.

Packer, C.; Fang, V.; Patil, S.; Lin, K.; Wooders, S.; and Gonzalez, J. 2023. MemGPT: towards LLMs as operating systems.

Park, J. S.; O'Brien, J.; Cai, C. J.; Morris, M. R.; Liang, P.; and Bernstein, M. S. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, 1–22.

Shinn, N.; Cassano, F.; Berman, E.; Gopinath, A.; Narasimhan, K.; and Yao, S. 2024. Reflexion: Language agents with verbal reinforcement learning, 2023. *URL https://arxiv. org/abs/2303.11366*, 8.

Tan, H.; Zhang, Z.; Cao, Z.; Li, R.; and Chen, X. 2026. DELTAMEM: Incremental Experience Memory for LLM Agents via Residual Trees. *arXiv preprint arXiv:2606.03083*.

Trivedi, H.; Balasubramanian, N.; Khot, T.; and Sabharwal, A. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, 10014–10037.

Uddin, M. N.; Shubham, K.; Blanco, E.; Baral, C.; and Wang, G. 2026. From recall to forgetting: Benchmarking long-term memory for personalized agents. *arXiv preprint arXiv:2604.20006*.

Wang, Z. Z.; Mao, J.; Fried, D.; and Neubig, G. 2024. Agent workflow memory. *arXiv preprint arXiv:2409.07429*.

Xiong, W.; Li, X.; Iyer, S.; Du, J.; Lewis, P.; Wang, W. Y.; Mehdad, Y.; Yih, S.; Riedel, S.; Kiela, D.; et al. 2020.

Answering complex open-domain questions with multi-hop dense retrieval. In *International Conference on Learning Representations*.

Yang, T.; Paul, S.; Srinivasan, V.; Kulkarni, V.; and Chappidi, S. 2026. TRUSTMEM: Learning Trustworthy Memory Consolidation for LLM Agents with Long-Term Memory. *arXiv preprint arXiv:2606.25161*.

Yang, Z.; Qi, P.; Zhang, S.; Bengio, Y.; Cohen, W.; Salakhutdinov, R.; and Manning, C. D. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, 2369–2380.

Yao, S.; Zhao, J.; Yu, D.; Shafran, I.; Narasimhan, K. R.; and Cao, Y. 2022. React: Synergizing reasoning and acting in language models. In *NeurIPS 2022 Foundation Models for Decision Making Workshop*.

Ye, C.; Liu, Y.; Wang, Y.; Yu, H.; Zhao, Y.; Liu, G.; McAuley, J.; and You, J. 2026. Auto-dreamer: Learning offline memory consolidation for language agents. *arXiv preprint arXiv:2605.20616*.

Zhao, A.; Huang, D.; Xu, Q.; Lin, M.; Liu, Y.-J.; and Huang, G. 2024. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 19632–19642.

Zhao, Y.; Yuan, B.; Huang, J.; Yuan, H.; Yu, Z.; Xu, H.; Hu, L.; Shankarampeta, A.; Huang, Z.; Ni, W.; et al. 2026. AMA-Bench: Evaluating long-horizon memory for agentic applications. In *Forty-third International Conference on Machine Learning*.

Zhong, W.; Guo, L.; Gao, Q.; Ye, H.; and Wang, Y. 2024. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, 19724–19731.

Zhong, Z.; Wu, Z.; Manning, C. D.; Potts, C.; and Chen, D. 2023. Mquake: Assessing knowledge editing in language models via multi-hop questions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 15686–15702.