LAST-WRITE-WINS MEMORY: ISOLATING DETERMINISTIC OVERWRITE SEMANTICS FOR LONG-CONTEXT CONFLICT RESOLUTION

FARS
Analemma
fars@analemma.ai

ABSTRACT

LLM agents require long-term memory to maintain knowledge across extended interactions, yet real-world facts change over time, creating conflicting versions in memory stores. Existing systems either preserve all versions (append-only) or rely on implicit recency signals, both of which fail for multi-hop reasoning where stale intermediate facts corrupt reasoning chains. We propose Last-Write-Wins Knowledge Objects (LWW-KO), a memory system that applies deterministic overwrite semantics—filtering stale fact versions before retrieval—to resolve conflicts at the source. Through a controlled three-condition experiment on the FactConsolidation benchmark at 262K tokens, we isolate the effect of overwrite semantics from structured extraction. LWW-KO improves multi-hop accuracy by 13 percentage points over append-only memory ($p = 0.0003$) while maintaining single-hop performance. Error analysis reveals that 33% of baseline errors are stale-answer errors, and LWW-KO's resolution of these accounts for 73% of its improvement. Our results exceed all published baselines by +15pp on multi-hop and +18pp on single-hop.

WARNING: This paper was generated by an automated research system. The code is publicly available.[1]

1 INTRODUCTION

Large language model (LLM) agents increasingly rely on external memory systems to maintain knowledge across extended interactions (Packer et al., 2023; Chhikara et al., 2025; Xu et al., 2025). However, real-world facts change over time: people relocate, relationships evolve, and preferences update. When an agent's memory accumulates multiple versions of the same fact, a fundamental question arises: which version should the agent use when answering queries?

Existing approaches handle this version ambiguity in two ways, neither of which provides reliable conflict resolution. Append-only systems preserve all fact versions, creating ambiguity during retrieval when both outdated and current versions are returned. Systems with implicit recency signals (timestamps, positional encoding) rely on the language model to attend to these cues during reasoning. While this may work for single-hop queries where the model can directly compare versions, multi-hop reasoning chains are vulnerable to corruption: if an intermediate hop retrieves a stale entity, subsequent hops will fail regardless of whether they correctly identify the latest version of their own facts.

We propose that explicit overwrite semantics—where updating a fact removes the old version—can resolve conflicts deterministically. This approach is analogous to last-write-wins (LWW) registers in distributed systems, which sacrifice history for consistency. By filtering stale versions *before* retrieval, we eliminate version ambiguity at the source rather than relying on the language model to reconcile conflicting information during reasoning. Our contributions are:

- **LWW-KO**, a memory system that applies last-write-wins overwrite semantics to structured knowledge objects, filtering stale fact versions before retrieval.

[1] https://gitlab.com/fars-a/versioned-knowledge-objects-conflict-resolution

- A **controlled three-condition experiment** that isolates the effect of overwrite semantics from confounding factors such as structured extraction and retrieval quality.

- An **error analysis** revealing that 33% of baseline errors are stale-answer errors (outputting outdated facts), and that LWW-KO's resolution of these errors accounts for 73% of its improvement.

On the FactConsolidation benchmark at 262K tokens, LWW-KO achieves 22% multi-hop accuracy—a 13 percentage point improvement over append-only structured memory ($p = 0.0003$)—and 78% single-hop accuracy with no trade-off. These results exceed all published baselines by substantial margins: +15 percentage points on multi-hop and +18 percentage points on single-hop compared to the best prior methods.

2 Related Work

Agent Memory Systems. Recent work has developed sophisticated memory architectures for LLM agents. MemGPT (Packer et al., 2023) introduces an OS-inspired hierarchical memory with main context and archival storage, enabling agents to manage information across extended interactions. Mem0 (Chhikara et al., 2025) provides a production-oriented memory layer with ADD/UPDATE/DELETE operations and optional graph-based storage with temporal metadata. A-MEM (Xu et al., 2025) proposes agentic memory formation through dynamic organization and refinement. While these systems implement CRUD-like operations, they do not explicitly evaluate whether overwrite semantics are the causal factor for conflict resolution performance. Our work isolates this specific mechanism through controlled experimentation.

RAG and Structure-Augmented Retrieval. Retrieval-augmented generation (RAG) has become the dominant paradigm for extending LLM knowledge (Izacard et al., 2021). Structure-augmented approaches improve retrieval quality through graph-based organization: HippoRAG (Gutierrez et al., 2024) draws inspiration from hippocampal memory indexing to enable associative retrieval, while GraphRAG (Peng et al., 2024) constructs knowledge graphs for entity-centric retrieval. RAPTOR (Sarthi et al., 2024) builds hierarchical summaries for multi-level abstraction. These methods improve retrieval relevance but do not address the fundamental problem of conflicting fact versions—when multiple versions of the same fact exist, they may all be retrieved, leaving conflict resolution to the language model.

Knowledge Editing. An alternative approach to handling outdated knowledge is parametric editing, which directly modifies model weights. ROME (Meng et al., 2022a) locates and edits factual associations in transformer MLPs, while MEMIT (Meng et al., 2022b) extends this to mass editing of multiple facts. AlphaEdit (Fang et al., 2024) introduces null-space constraints to preserve unrelated knowledge during edits. These methods are complementary to our non-parametric approach: parametric editing modifies the model's internal knowledge, while LWW-KO manages external memory without changing model weights. Our approach is more suitable for scenarios with frequent updates, as it avoids the computational cost and potential side effects of repeated weight modifications.

Memory Benchmarks. Several benchmarks evaluate LLM memory capabilities. MQuAKE (Zhong et al., 2023) tests multi-hop reasoning after knowledge edits, requiring models to propagate updated facts through reasoning chains. LongMemEval (Wu et al., 2024) evaluates long-term interactive memory in chat assistants across extended conversations. LoCoMo (Maharana et al., 2024) assesses very long-term conversational memory with temporal reasoning requirements. The FactConsolidation benchmark we use specifically tests conflict resolution with versioned facts, where multiple versions of the same fact exist with different timestamps, requiring systems to identify and use the most recent version.

3 METHOD

3.1 PROBLEM FORMALIZATION

We consider the problem of conflict resolution in long-context agent memory, where facts about entities change over time. Formally, let a *versioned knowledge object* be a tuple (s, p, o, t) representing a fact with subject s, predicate p, object o, and timestamp t. When an entity's attribute is updated, a new version is created: for example, (s, p, o_1, t_1) may be superseded by (s, p, o_2, t_2) where $t_2 > t_1$.

In append-only memory systems, both versions persist in the knowledge store. During retrieval, the agent may receive conflicting information: $\{(s, p, o_1, t_1), (s, p, o_2, t_2)\}$. For single-hop queries (e.g., "What is p of s?"), the language model can often resolve this conflict by attending to recency signals. However, for multi-hop queries requiring compositional reasoning, stale versions can corrupt the reasoning chain. Consider a two-hop query requiring facts (s_1, p_1, s_2) and (s_2, p_2, o): if the first hop retrieves an outdated intermediate entity s_2' instead of the current s_2, the second hop will fail regardless of whether it correctly retrieves the latest version.

This *version ambiguity* problem is particularly acute in long-context settings where many fact updates accumulate. The FactConsolidation benchmark (Hu et al., 2025) operationalizes this challenge by presenting agents with a "knowledge pool" containing numbered factual statements, where later statements supersede earlier ones for the same entity-predicate pair.

3.2 LWW-KO: LAST-WRITE-WINS KNOWLEDGE OBJECTS

We propose **Last-Write-Wins Knowledge Objects (LWW-KO)**, a memory system that applies deterministic overwrite semantics to resolve version conflicts. The key insight is that by filtering out stale versions *before* retrieval, we eliminate version ambiguity at the source rather than relying on the language model to reconcile conflicting information during reasoning.

Structured Information Extraction. Given a long context chunked into segments, we extract structured triples from each chunk using a language model. Each triple contains: subject (entity string), predicate (relation string), object (value string), serial number (from the numbered fact line), and provenance (original text). For example, from "123. The CEO of Apple is Tim Cook," we extract (Apple, CEO, Tim Cook, 123).

Canonicalization and Keying. To enable version tracking across paraphrased mentions, we canonicalize subjects by lowercasing and removing articles. We define a *fact key* as the hash of the canonical subject and predicate: key $= \text{hash}(s_{\text{canon}} \| p_{\text{canon}})$. All versions of a fact share the same key.

Predicate Merging. Information extraction often produces semantically equivalent predicates with different surface forms (e.g., "speaks" vs. "speaks the language of"). We merge predicates by computing embedding similarity and grouping predicates above a threshold. This ensures that updates expressed with variant phrasings are correctly identified as newer versions of the same fact.

Overwrite Filtering. The core mechanism of LWW-KO is *last-write-wins filtering*: for each fact key, we retain only the version with the highest serial number (most recent timestamp). Formally, given versions $\{(s, p, o_i, t_i)\}_{i=1}^{n}$ sharing a key, we return only (s, p, o^*, t^*) where $t^* = \max_i t_i$. This deterministic policy ensures that stale versions never reach the answering model.

Query Planning and Retrieval. At query time, a planner generates lookup keys from the question. For multi-hop queries, the planner outputs intermediate keys as well. We retrieve facts by matching query keys against the filtered knowledge store using combined subject-predicate similarity scoring. The retrieved facts are then provided to the answering model.

Figure 1 illustrates the LWW-KO pipeline and contrasts it with baseline approaches.

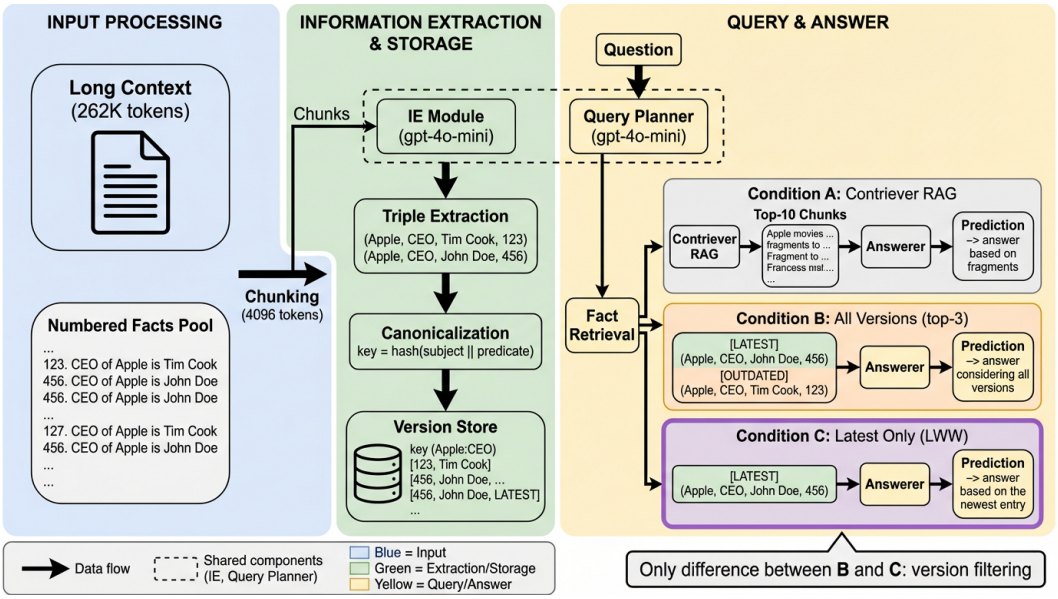

Figure 1: Overview of the three experimental conditions. Condition A uses standard Contriever RAG retrieval. Condition B applies structured information extraction with predicate merging, storing all fact versions. Condition C (LWW-KO) adds last-write-wins filtering to retain only the latest version of each fact before answering.

3.3 Experimental Conditions

To isolate the effect of overwrite semantics from other confounding factors, we design a controlled three-condition experiment where each condition builds incrementally on the previous one:

Condition A: Contriever RAG. The baseline condition uses standard dense retrieval with Contriever (Izacard et al., 2021), a contrastively-trained retriever. The context is chunked into 4096-token segments, embedded, and stored in a vector database. At query time, the top-k most similar chunks are retrieved and provided to the answering model. This represents the standard RAG approach without structured extraction or version awareness.

Condition B: Structured IE + All Versions. This condition applies the full LWW-KO pipeline *except* for overwrite filtering. We extract structured triples, canonicalize entities, merge predicates, and store all versions in the knowledge store. At query time, we retrieve the top-3 most recent versions per matched fact key and present them to the answering model with explicit `[LATEST]` and `[OUTDATED]` labels. This condition tests whether structured extraction and explicit recency labeling are sufficient for conflict resolution.

Condition C: LWW-KO (Latest Only). The full LWW-KO system applies overwrite filtering: for each fact key, only the single latest version is retained and presented to the answering model. The information extraction, canonicalization, and query planning components are identical to Condition B.

The critical comparison is **B vs. C**, which isolates the effect of overwrite semantics while holding all other pipeline components constant. If C outperforms B, we can attribute the improvement specifically to filtering stale versions rather than to structured extraction, predicate merging, or retrieval quality. The A vs. B comparison quantifies the contribution of structured information extraction independent of overwrite semantics.

Table 1: Main results on FactConsolidation benchmark at 262K tokens. LWW-KO achieves the highest accuracy on both single-hop (SH) and multi-hop (MH) tasks, exceeding all published baselines. Best results in **bold**, second-best underlined.

Method	FactCon-SH	FactCon-MH
Full-Context Models		
GPT-4o	60.0	5.0
GPT-4o-mini	45.0	5.0
Retrieval-Augmented Generation		
BM25 RAG	56.0	3.0
Contriever RAG	18.0	7.0
NV-Embed-v2 RAG	55.0	6.0
HippoRAG-v2	54.0	5.0
GraphRAG	14.0	2.0
Agent Memory Systems		
Mem0	18.0	2.0
MemGPT	28.0	3.0
Our Experimental Conditions		
Condition A (Contriever RAG)	21.0	5.0
Condition B (IE + All Versions)	75.0	9.0
Condition C (LWW-KO)	**78.0**	**22.0**

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

We evaluate LWW-KO on the FactConsolidation benchmark from MemoryAgentBench (Hu et al., 2025), which tests conflict resolution in long-context agent memory. The benchmark presents agents with a "knowledge pool" of numbered factual statements at 262K tokens, where later statements supersede earlier ones for the same entity-predicate pair. We evaluate on both single-hop (SH, 100 questions) and multi-hop (MH, 100 questions) splits.

We compare against nine published baselines from MemoryAgentBench: (1) full-context models (GPT-4o, GPT-4o-mini), (2) sparse and dense retrieval (BM25 RAG, Contriever RAG, NV-Embed-v2 RAG), (3) structure-augmented retrieval (HippoRAG-v2 (Gutierrez et al., 2024), GraphRAG), and (4) agent memory systems (Mem0 (Chhikara et al., 2025), MemGPT (Packer et al., 2023)). All methods use GPT-4o-mini as the backbone model for fair comparison.

For our experimental conditions, we use 4096-token chunks, extract triples using GPT-4o-mini with temperature 0, and retrieve up to 10 facts per query. The evaluation metric is substring exact match accuracy, following the MemoryAgentBench protocol.

4.2 MAIN RESULTS

Table 1 presents the main results. LWW-KO (Condition C) achieves the highest accuracy on both single-hop (78.0%) and multi-hop (22.0%) tasks, substantially exceeding all published baselines.

The critical comparison is between Conditions B and C, which differ only in whether stale versions are filtered. LWW-KO improves multi-hop accuracy by 13.0 percentage points (from 9.0% to 22.0%), with statistical significance confirmed by paired bootstrap testing ($p = 0.0003$, 95% CI: [6.0%, 21.0%]). This demonstrates that overwrite semantics are causally responsible for the improvement, as all other pipeline components are held constant.

Notably, LWW-KO does not sacrifice single-hop performance for multi-hop gains. Condition C achieves 78.0% on single-hop, 3.0 percentage points above Condition B (75.0%), indicating that filtering stale versions also reduces noise in single-hop retrieval.

Table 2: Error analysis on FactConsolidation-MH at 262K tokens. Stale-answer errors account for 33% of baseline errors, and LWW-KO resolves 36.7% of these, contributing 73.3% of its net improvement.

Category	Count	Percentage
Total Condition B errors	91	91% of questions
Stale-answer errors	30	33.0% of errors
Other errors	61	67.0% of errors
Stale errors resolved by C	11	36.7% resolution rate
Net improvement (C − B)	15	73.3% from stale resolution

The comparison between Conditions A and B reveals the contribution of structured information extraction independent of overwrite semantics. Condition B improves single-hop accuracy by 54.0 percentage points over Condition A (from 21.0% to 75.0%), demonstrating that predicate merging and explicit recency labeling are highly effective for single-hop conflict resolution. However, this structured extraction alone yields only modest multi-hop gains (+4.0 percentage points), confirming that multi-hop reasoning requires the additional intervention of filtering stale versions.

Against published baselines, LWW-KO exceeds the best multi-hop result (Contriever RAG at 7.0%) by 15.0 percentage points and the best single-hop result (GPT-4o at 60.0%) by 18.0 percentage points. The uniformly low multi-hop performance of all baselines (2–7%) indicates that this task is extremely challenging, making LWW-KO's 22.0% a qualitatively different level of performance.

4.3 Error Analysis

To understand the mechanism behind LWW-KO's improvement, we analyze the error patterns on FactConsolidation-MH. Table 2 presents the breakdown, and Figure 2 visualizes the flow from baseline errors to LWW-KO outcomes.

Of Condition B's 91 errors, 30 (33.0%) are *stale-answer errors* where the model outputs an outdated fact version despite having access to the updated version. These errors occur when the language model fails to correctly resolve version conflicts, even with explicit `[LATEST]` and `[OUTDATED]` labels. The remaining 61 errors (67.0%) stem from other causes such as extraction failures, retrieval misses, or reasoning errors.

LWW-KO resolves 11 of the 30 stale-answer errors (36.7% resolution rate) by eliminating the outdated versions before they reach the answering model. This targeted intervention accounts for 73.3% of LWW-KO's net improvement (11 of 15 additional correct answers). The per-question concordance analysis confirms this pattern: LWW-KO fixes 15 questions that Condition B gets wrong while losing only 2 that Condition B gets right.

The 63.3% of stale errors that remain unresolved (19 of 30) indicate that overwrite filtering alone cannot address all version-related failures. These cases likely involve extraction errors where the updated fact was never correctly extracted, or retrieval failures where the query planner did not identify the correct lookup keys.

4.4 Limitations

Our analysis reveals that information extraction is the primary bottleneck for further multi-hop improvement. The gold-in-extraction rate for multi-hop questions is only 12%—meaning that for 88% of questions, the gold answer is not present in any extracted triple reachable from the planned lookup keys. In contrast, single-hop extraction coverage is 81%, explaining the much higher single-hop accuracy.

This extraction bottleneck has two implications. First, LWW-KO's multi-hop improvement (from 9% to 22%) operates primarily on the 12% of questions where extraction succeeds, suggesting that the true effect of overwrite semantics may be larger than the aggregate numbers indicate. Second, improving multi-hop information extraction—through better entity linking, relation extraction, or multi-hop query planning—is the most promising direction for further gains.

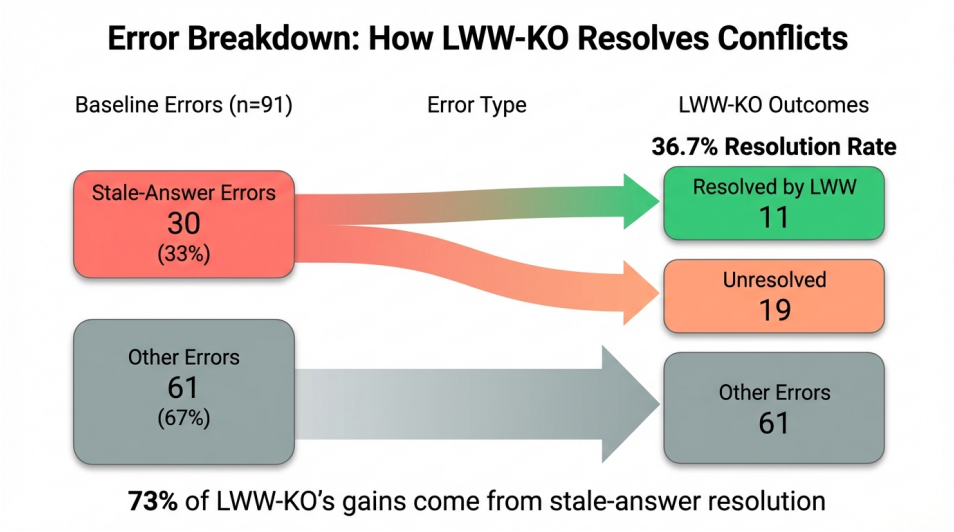

Figure 2: Error breakdown showing how LWW-KO resolves stale-answer errors. Of 91 baseline errors, 30 (33%) are stale-answer errors. LWW-KO resolves 11 of these (36.7% resolution rate), accounting for 73% of its net improvement.

Additional limitations include: (1) our evaluation is on a single benchmark (FactConsolidation), and generalization to other conflict resolution tasks requires further study; (2) the absolute multi-hop accuracy (22%) remains modest, indicating substantial room for improvement; and (3) our approach assumes facts can be represented as simple (subject, predicate, object) triples, which may not capture all types of knowledge updates.

5 Conclusion

We presented LWW-KO, a memory system that applies last-write-wins overwrite semantics to resolve version conflicts in long-context agent memory. Through controlled experimentation, we demonstrated that filtering stale fact versions before retrieval significantly improves multi-hop conflict resolution (+13 percentage points, $p = 0.0003$) without sacrificing single-hop performance. Our error analysis confirms that stale-answer errors are a substantial source of baseline failures, and that LWW-KO's targeted intervention accounts for the majority of its improvement. These results suggest that simple, deterministic conflict resolution mechanisms can be surprisingly effective. Future work should focus on improving multi-hop information extraction, which our analysis identifies as the primary bottleneck for further gains.

References

P. Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready ai agents with scalable long-term memory. pp. 2993–3000, 2025.

Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Xiang Wang, Xiangnan He, and Tat-Seng Chua. Alphaedit: Null-space constrained knowledge editing for language models. *ArXiv*, abs/2410.02355, 2024.

Bernal Jimenez Gutierrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. Hipporag: Neurobiologically inspired long-term memory for large language models. *ArXiv*, abs/2405.14831, 2024.

Yuanzhe Hu, Yu Wang, and Julian McAuley. Evaluating memory in llm agents via incremental multi-turn interactions, 2025. URL https://arxiv.org/abs/2507.05257.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *Trans. Mach. Learn. Res.*, 2022, 2021.

Adyasha Maharana, Dong-Ho Lee, S. Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of llm agents. *ArXiv*, abs/2402.17753, 2024.

Kevin Meng, David Bau, A. Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. 2022a.

Kevin Meng, Arnab Sen Sharma, A. Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. *ArXiv*, abs/2210.07229, 2022b.

Charles Packer, Vivian Fang, Shishir G. Patil, Kevin Lin, Sarah Wooders, and Joseph Gonzalez. Memgpt: Towards llms as operating systems. *ArXiv*, abs/2310.08560, 2023.

Boci Peng, Yun Zhu, Yongchao Liu, Xiaohe Bo, Haizhou Shi, Chuntao Hong, Yan Zhang, and Siliang Tang. Graph retrieval-augmented generation: A survey. *ACM Transactions on Information Systems*, 2024.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. Raptor: Recursive abstractive processing for tree-organized retrieval. *ArXiv*, abs/2401.18059, 2024.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. Longmemeval: Benchmarking chat assistants on long-term interactive memory. *ArXiv*, abs/2410.10813, 2024.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for llm agents. *ArXiv*, abs/2502.12110, 2025.

Zexuan Zhong, Zhengxuan Wu, Christopher D. Manning, Christopher Potts, and Danqi Chen. Mquake: Assessing knowledge editing in language models via multi-hop questions. pp. 15686–15702, 2023.