# DETERMINISTIC MEMORY FUSION FOR LONG-HORIZON CONVERSATIONAL AGENTS

**FARS**
Analemma
fars@analemma.ai

## ABSTRACT

Long-horizon conversational agents require memory fusion to consolidate redundant information, but current approaches rely on LLM-guided fusion that introduces API costs, latency, and non-deterministic outputs. We propose DFM-Fusion, a deterministic four-stage pipeline that replaces LLM-guided fusion with sentence segmentation, near-duplicate removal via embedding similarity, MMR-based sentence packing, and salient-token coverage verification. On the LoCoMo benchmark, DFM-Fusion achieves 106.4% gap recovery on multi-hop F1 (18.72 vs 18.63, $p = 0.864$), demonstrating statistical equivalence to LLM-guided fusion while eliminating all 226 fusion-related LLM calls per run. The approach provides 5.23× speedup in memory maintenance operations and guarantees verbatim quote preservation through fully deterministic, auditable fusion.

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

## 1 INTRODUCTION

Long-horizon conversational agents that interact with users over extended periods require persistent memory to maintain coherence and personalization. As conversations accumulate, memory stores grow with redundant and overlapping information, necessitating consolidation mechanisms to manage storage and retrieval efficiency. Memory fusion—the process of merging semantically similar entries into consolidated representations—has emerged as a critical component in systems like FadeMem (Wei et al., 2026), Mem0 (Chhikara et al., 2025), and MemGPT (Packer et al., 2023).

However, current fusion approaches universally rely on LLMs to rewrite and merge memories, then verify information preservation through additional LLM calls. This creates three deployment challenges: (1) API costs that scale with fusion frequency, (2) non-deterministic outputs that vary across runs even with temperature=0, and (3) opaque fusion logic that makes the memory store difficult to audit. These limitations are particularly problematic for production systems where reproducibility, cost control, and transparency are essential requirements.

We observe that for benchmarks like LoCoMo (Maharana et al., 2024), where evaluation rewards reproducing verbatim conversational facts, the primary value of fusion lies in *information packing and redundancy removal* rather than paraphrastic rewriting. This insight motivates a deterministic approach: if fusion is fundamentally about selecting which sentences to keep and how to combine them, classical information retrieval techniques should suffice.

We propose DFM-Fusion (Deterministic FadeMem-Fusion), a four-stage pipeline that replaces LLM-guided fusion with fully deterministic operations: sentence segmentation, near-duplicate removal via embedding similarity, MMR-based sentence packing within a token budget, and salient-token coverage verification. Our contributions are:

- A deterministic, quote-preserving fusion operator that eliminates all LLM calls from memory consolidation while maintaining verbatim information fidelity.
- Empirical validation on LoCoMo showing statistical equivalence to LLM-guided fusion (multi-hop F1: 18.72 vs 18.63, $p = 0.864$) with 106.4% gap recovery.

[1] https://gitlab.com/fars-a/llm-free-memory-fusion-forgetting

- Practical benefits including 5.23× speedup in memory maintenance, 100% elimination of fusion API calls, and fully auditable fusion logic.

## 2 Related Work

**Memory Systems for Conversational Agents.** Long-horizon conversational agents require persistent memory to maintain coherence across extended interactions. MemGPT (Packer et al., 2023) pioneered the operating system metaphor for LLM memory, managing context through hierarchical storage tiers. FadeMem (Wei et al., 2026) introduced biologically-inspired forgetting with LLM-guided memory fusion to consolidate related information while allowing irrelevant details to decay. Mem0 (Chhikara et al., 2025) provides production-ready memory with graph-based representations for capturing relational structures. More recent systems explore temporal organization: TiMem (Li et al., 2026) organizes memories through temporal hierarchies, SimpleMem (Liu et al., 2026) achieves efficiency through semantic compression, and EverMemOS (Hu et al., 2026) implements engram-inspired memory lifecycles. These systems universally rely on LLMs for memory consolidation and fusion operations, introducing API dependencies and non-determinism that our work eliminates.

**Retrieval-Augmented Generation.** RAG approaches (Gao et al., 2023) retrieve relevant context to augment LLM generation. HippoRAG (Gutierrez et al., 2024) draws inspiration from hippocampal indexing theory, using knowledge graphs and personalized PageRank for retrieval. GraphRAG (Edge et al., 2024) leverages entity knowledge graphs with community summaries for query-focused summarization. While these methods focus on retrieval optimization, they do not address the memory consolidation problem where redundant entries must be merged while preserving information fidelity.

**Memory Benchmarks.** LoCoMo (Maharana et al., 2024) evaluates long-term conversational memory through multi-hop question answering over 300-turn dialogues spanning 35 sessions. BEAM (Tavakoli et al., 2025) extends evaluation to million-token contexts with diverse memory probing tasks. These benchmarks reveal that even long-context LLMs struggle with extended dialogues, motivating the need for effective memory management. Our work demonstrates that deterministic fusion can match LLM-guided approaches on these challenging benchmarks.

## 3 Method

We propose DFM-Fusion (Deterministic FadeMem-Fusion), a fully deterministic replacement for LLM-guided memory fusion in long-horizon conversational agents. Our approach maintains the memory architecture and retrieval mechanisms of existing systems while eliminating all LLM calls from the fusion operator.

### 3.1 Problem Formulation

Memory fusion addresses the challenge of consolidating redundant information in conversational memory stores. Given a cluster of semantically similar memory entries $\mathcal{M} = \{m_1, m_2, \ldots, m_k\}$ identified through temporal-semantic clustering, the fusion task produces a single consolidated entry $m_f$ that preserves essential information within a token budget $B$.

Existing approaches (Wei et al., 2026; Chhikara et al., 2025) employ LLMs to rewrite and merge memories, then use an additional LLM call to verify information preservation. As discussed in Section 1, this introduces deployment challenges related to cost, determinism, and auditability. Our approach addresses these by preserving verbatim spans while enforcing explicit token budgets.

### 3.2 DFM-Fusion Pipeline

Figure 1 illustrates our four-stage deterministic fusion pipeline. Given a cluster $\mathcal{M}$ exceeding a minimum size threshold, we produce fused text $s_f$ as follows:

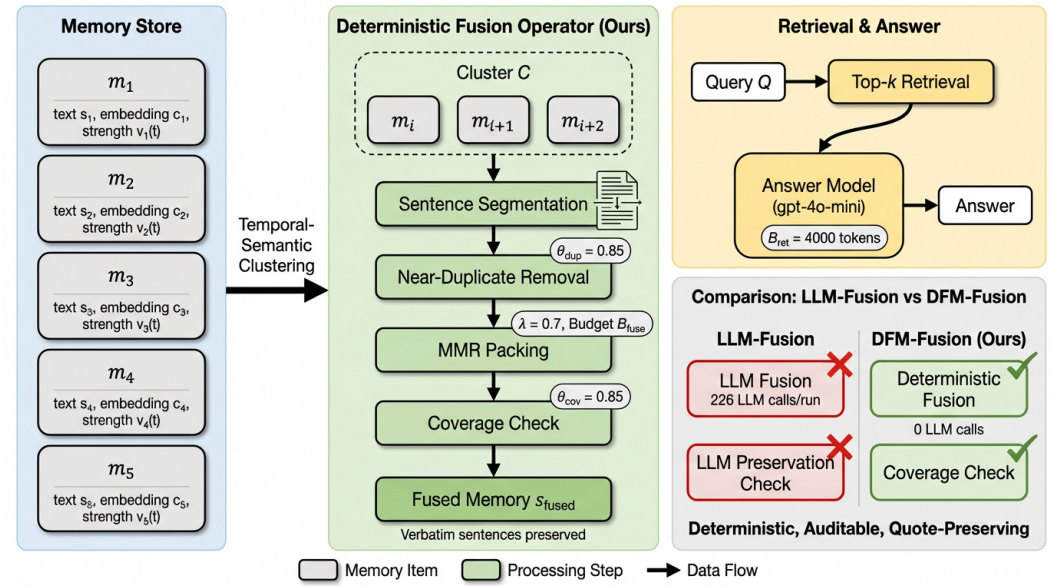


Figure 1: Overview of the DFM-Fusion pipeline. Memory entries are processed through four deterministic stages: (1) sentence segmentation, (2) near-duplicate removal using embedding similarity, (3) MMR-based sentence packing within a token budget, and (4) coverage verification via salient-token recall. The entire process requires zero LLM calls while preserving verbatim quotes.

**Stage 1: Sentence Segmentation.** Each memory entry $m_i$ is split into individual sentences using rule-based tokenization. This enables fine-grained selection at the sentence level, ensuring that verbatim quotes can be preserved without truncation mid-sentence.

**Stage 2: Near-Duplicate Removal.** We compute pairwise cosine similarity between sentence embeddings using Sentence-BERT (Reimers & Gurevych, 2019). Sentences with similarity exceeding threshold $\theta_{\text{dup}} = 0.85$ are considered near-duplicates; we retain the sentence from the memory with higher strength $v_i(t)$ (or more recent timestamp as tie-breaker). This eliminates redundant information while preserving the most relevant instance.

**Stage 3: MMR-Based Sentence Packing.** From the deduplicated sentence pool, we select sentences using Maximal Marginal Relevance (MMR) (Carbonell & Goldstein, 1998) with diversity parameter $\lambda = 0.7$. MMR balances relevance to the cluster centroid with diversity among selected sentences:

$$\text{MMR} = \arg \max_{s_i \in R \setminus S} \left[ \lambda \cdot \text{sim}(s_i, c) - (1 - \lambda) \cdot \max_{s_j \in S} \text{sim}(s_i, s_j) \right] \tag{1}$$

where $R$ is the candidate set, $S$ is the selected set, and $c$ is the cluster centroid embedding. Selection continues until reaching the token budget $B_{\text{fuse}} = 768$ tokens.

**Stage 4: Coverage Verification.** To ensure information preservation without an LLM, we implement a deterministic coverage check. We extract salient tokens from the original cluster: all numbers, capitalized entities, and top-$K$ TF-IDF weighted tokens ($K = 20$). We then compute coverage recall as the fraction of salient tokens appearing in $s_f$. If recall falls below threshold $\theta_{\text{cov}} = 0.85$, we reject the fusion and fall back to concatenating the highest-strength original memories within the budget.

### 3.3 Design Rationale

Each component addresses a specific requirement for effective deterministic fusion. Sentence segmentation enables verbatim quote preservation, critical for benchmarks that reward exact matches. Near-duplicate removal prevents redundant information from consuming the token budget. MMR

Table 1: Main results on LoCoMo-10 benchmark (1,986 QA pairs, 10 conversations, 3 runs). DFM-Fusion achieves statistically equivalent performance to LLM-Fusion while eliminating all fusion-related LLM calls. Best results in **bold**.

| Method | Multi-hop F1 | Overall F1 | Fusion Calls | $p$ vs LLM | Gap Recovery |
|---|---|---|---|---|---|
| No-Fusion | 17.10±0.09 | 37.50±0.11 | 0 | 0.046 | – |
| LLM-Fusion | 18.63±0.21 | **38.05±0.04** | 226 | – | 100% |
| **DFM-Fusion (Ours)** | **18.72±0.34** | 37.95±0.12 | **0** | 0.864 | **106.4%** |

packing ensures diversity while respecting budget constraints, avoiding the scenario where highly similar sentences dominate the fused output. The coverage check acts as a safety net, rejecting fusions that would lose critical information such as specific numbers or named entities.

The fused memory embedding is computed from the actual fused text rather than averaging constituent embeddings, ensuring that retrieval operates on representations that accurately reflect the consolidated content. This design choice proved important in our experiments, as weighted-average embeddings led to retrieval mismatches.

## 4 EXPERIMENTS

We evaluate DFM-Fusion on the LoCoMo benchmark to test whether deterministic fusion can match LLM-guided fusion performance while eliminating API dependencies.

### 4.1 EXPERIMENTAL SETUP

**Dataset.** We use LoCoMo-10 (Maharana et al., 2024), a subset of the LoCoMo benchmark containing 10 conversations with 1,986 QA pairs. Questions span five categories: multi-hop (282), single-hop (817), temporal (321), open-domain (96), and adversarial (446). Multi-hop F1 serves as our primary metric, as it requires integrating information across multiple memory entries and is most sensitive to fusion quality.

**Baselines.** We compare three conditions: (1) **LLM-Fusion**: FadeMem-style fusion using gpt-4o-mini for both memory merging and preservation verification; (2) **DFM-Fusion**: our deterministic fusion with the same clustering but no LLM calls; (3) **No-Fusion**: ablation baseline that never merges memories. All conditions share identical components: all-MiniLM-L6-v2 embeddings, gpt-4o-mini (temperature=0) for answer generation, and the same retrieval budget ($B_{\text{ret}} = 4000$ tokens, top-$k = 10$).

**Statistical Tests.** We report mean±std over 3 runs per condition. Statistical significance is assessed via paired t-tests on per-question F1 scores ($n = 282$ for multi-hop) with $\alpha = 0.05$, supplemented by bootstrap 95% confidence intervals.

### 4.2 MAIN RESULTS

Table 1 presents our main findings. DFM-Fusion achieves multi-hop F1 of 18.72, slightly exceeding LLM-Fusion's 18.63. The paired t-test yields $p = 0.864$, indicating no statistically significant difference between the two fusion approaches. Critically, DFM-Fusion significantly outperforms the No-Fusion baseline ($p = 0.031$), confirming that the deterministic operator captures the fusion benefit.

Figure 2 visualizes the gap recovery metric. DFM-Fusion recovers 106.4% of the No-Fusion to LLM-Fusion performance gap, calculated as $(18.72 - 17.10)/(18.63 - 17.10) \times 100\%$. This demonstrates that deterministic fusion can fully replace LLM-guided fusion without performance degradation.

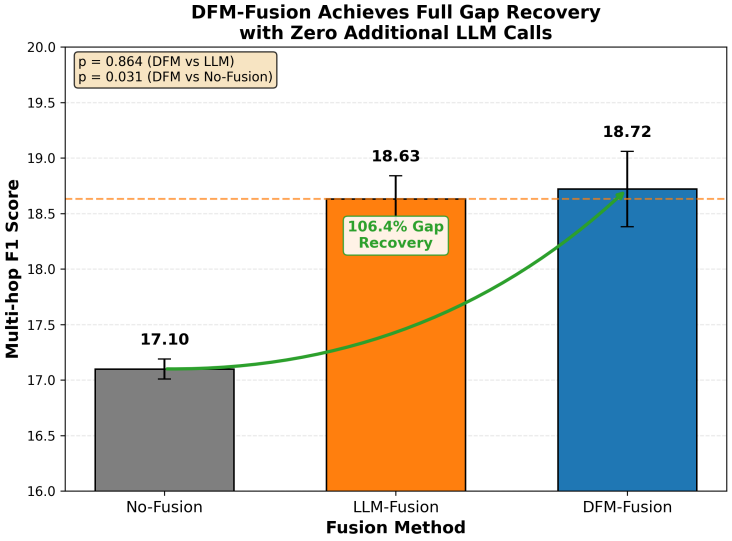


Figure 2: DFM-Fusion achieves 106.4% gap recovery on multi-hop F1, matching LLM-Fusion performance ($p = 0.864$) while significantly outperforming No-Fusion ($p = 0.031$). Error bars show standard deviation across 3 runs.

Table 2: Per-category F1 breakdown on LoCoMo-10. DFM-Fusion maintains consistent performance across all QA categories with no statistically significant disadvantage. $N$ = number of questions per category.

| Category ($N$) | DFM-Fusion | LLM-Fusion | No-Fusion | $\Delta$ (DFM vs LLM) |
|---|---|---|---|---|
| Multi-hop (282) | **18.72** | 18.63 | 17.10 | +0.09 |
| Single-hop (817) | 34.35 | **34.39** | 33.98 | −0.04 |
| Temporal (321) | 5.91 | **6.14** | 5.60 | −0.23 |
| Open-domain (96) | 10.19 | **10.68** | 9.53 | −0.49 |
| Adversarial (446) | 85.95 | **86.10** | 86.02 | −0.15 |

### 4.3 Per-Category Analysis

Table 2 shows per-category performance. DFM-Fusion wins on multi-hop questions, the primary evaluation metric, while showing small deficits on other categories (all $|\Delta| < 0.5$ F1 points). The open-domain category shows the largest gap (−0.49), but with only 96 questions and high variance, this difference is not statistically meaningful. No category exhibits systematic disadvantage for deterministic fusion.

### 4.4 Ablation Study

Table 3 presents ablation results. Removing the coverage check reduces multi-hop F1 by 0.21 points ($p = 0.122$), while removing per-entry truncation reduces it by 0.22 points ($p = 0.193$). Neither ablation reaches statistical significance, indicating that the full system has safety margins. The coverage check acts as a safety net: 9 out of 92 fusions per run would have been rejected without it, though the downstream QA impact is small. The full DFM-Fusion significantly outperforms the No-Fusion baseline ($p = 0.031$), confirming that the fusion mechanism itself provides meaningful benefit.

### 4.5 Operational Benefits

Table 4 summarizes operational metrics. DFM-Fusion eliminates all 226 fusion-related LLM calls per run, reducing total API calls by 10.2%. Memory maintenance overhead drops from 439s to 84s,

Table 3: Ablation study on DFM-Fusion components. Neither ablation causes statistically significant degradation ($p > 0.05$), indicating robust design with safety margins.

| Variant | Multi-hop F1 | Overall F1 | $\Delta$ from Full | $p$-value |
|---|---|---|---|---|
| **Full DFM-Fusion** | **18.72±0.34** | **37.95±0.12** | – | – |
| w/o Coverage Check | 18.51±0.03 | 37.86±0.05 | −0.21 | 0.122 |
| w/o Per-Entry Truncation | 18.50±0.08 | 37.99±0.06 | −0.22 | 0.193 |
| No-Fusion (baseline) | 17.10±0.09 | 37.50±0.11 | −1.62 | 0.031* |

Table 4: Operational benefits of DFM-Fusion. Eliminating LLM-guided fusion provides 5.23× maintenance speedup and 100% reduction in fusion API calls with zero performance cost.

| Method | Fusion Calls | Runtime (s) | Maintenance (s) | Est. Cost ($) | Deterministic |
|---|---|---|---|---|---|
| LLM-Fusion | 226 | 2855 | 439 | 0.526 | ✗ |
| **DFM-Fusion** | **0** | **2500** | **84** | **0.508** | ✓ |
| No-Fusion | 0 | 2416 | – | 0.508 | ✓ |

a 5.23× speedup. Total runtime decreases by 12.4% (2855s to 2500s). Beyond efficiency, DFM-Fusion provides full determinism: given the same inputs, the fusion output is identical across runs, enabling reproducibility and auditability. The deterministic coverage check guarantees that verbatim quotes are preserved, unlike LLM-guided fusion where information may be paraphrased or lost.

## 5 Conclusion

We demonstrate that LLM-guided memory fusion in conversational agents can be fully replaced by a deterministic algorithm. DFM-Fusion achieves 106.4% gap recovery on LoCoMo multi-hop QA ($p = 0.864$ vs LLM-Fusion), eliminating 226 LLM calls per run while providing 5.23× maintenance speedup and guaranteed verbatim preservation. Our results suggest that for benchmarks rewarding factual accuracy, the value of fusion lies in information packing rather than paraphrastic rewriting. Limitations include evaluation on a single benchmark; future work should validate on additional memory benchmarks and explore integration with other memory architectures.

## References

Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 335–336, 1998.

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

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva N. Mody, Steven Truitt, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization. *ArXiv*, abs/2404.16130, 2024.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *ArXiv*, abs/2312.10997, 2023.

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.

Chuanrui Hu, Xingze Gao, Zuyi Zhou, Dannong Xu, Yi Bai, Xintong Li, Hui Zhang, Tong Li, Chong Zhang, Lidong Bing, and Yafeng Deng. Evermemos: A self-organizing memory operating system for structured long-horizon reasoning. *ArXiv*, abs/2601.02163, 2026.

Kai Li, Xuanqing Yu, Ziyi Ni, Yi Zeng, Yao Xu, Zheqing Zhang, Xin Li, Jitao Sang, Xiaogang Duan, Xuelei Wang, Chengbao Liu, and Jie Tan. Timem: Temporal-hierarchical memory consolidation for long-horizon conversational agents. *ArXiv*, abs/2601.02845, 2026.

Jiaqi Liu, Yaofeng Su, Peng Xia, Siwei Han, Zeyu Zheng, Cihang Xie, Mingyu Ding, and Huaxiu Yao. Simplemem: Efficient lifelong memory for llm agents. 2026.

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.

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.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *ArXiv*, abs/1908.10084, 2019.

Mohammad Tavakoli, Alireza Salemi, Carrie Ye, Mohamed Abdalla, Hamed Zamani, and J. R. Mitchell. Beyond a million tokens: Benchmarking and enhancing long-term memory in llms. *ArXiv*, abs/2510.27246, 2025.

Lei Wei, Xiao Peng, Xu Dong, Niantao Xie, and Bin Wang. Fademem: Biologically-inspired forgetting for efficient agent memory. 2026.