SELF-ANCHORED TEMPORAL FILTERING FOR LLM-FREE TEMPORAL-AWARE MEMORY RETRIEVAL

FARS
Analemma
fars@analemma.ai

ABSTRACT

Long-term conversational memory systems require temporal awareness to retrieve contextually relevant information from past interactions. Current approaches either ignore temporal signals (pure dense retrieval) or require expensive LLM calls to extract time constraints from queries. We propose Self-Anchored Temporal Filtering (SATF), which infers temporal relevance from the timestamp distribution of initial retrieval results using multi-peak Gaussian kernels weighted by reciprocal rank. SATF soft-boosts temporally coherent items without hard filtering, requiring zero LLM calls. On LongMemEval, SATF achieves +16.9% relative NDCG@10 improvement on temporal reasoning queries (0.584→0.683) while outperforming GPT-4o time-range filtering across all metrics with zero API cost. SATF improves all question types without degrading non-temporal queries, demonstrating that temporal signals embedded in retrieval distributions can be effectively exploited for ranking.

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

1 INTRODUCTION

Long-term conversational memory is essential for AI assistants to maintain coherent, personalized interactions across extended time periods. As user-assistant interaction histories grow beyond the context window of language models, external memory systems with retrieval capabilities become necessary (Packer et al., 2023; Chhikara et al., 2025). A critical challenge in these systems is temporal awareness: users frequently ask about "recent" events, "last week's" discussions, or information that has been updated over time. Without temporal reasoning, retrieval systems may return semantically similar but temporally incorrect results, such as restaurant recommendations from months ago when the user asked about "last weekend."

Current approaches to temporal-aware memory retrieval face a fundamental trade-off. Pure dense retrieval (Karpukhin et al., 2020) ignores temporal signals entirely, relying solely on semantic similarity. LLM-based temporal filtering, as proposed in LongMemEval (Wu et al., 2024), uses a strong language model (e.g., GPT-4o) to extract time ranges from queries and filter candidates accordingly. While effective, this approach incurs significant API costs (one LLM call per query) and can over-filter, reducing recall when the extracted time range is too narrow.

We observe that the timestamp distribution of top-ranked retrieval results contains implicit temporal signals that can be exploited without explicit temporal reasoning. If a query has temporal intent, semantically relevant items will naturally cluster around specific time periods in the initial retrieval results. This "self-anchored" signal—derived from the retrieval distribution itself—can identify temporally relevant time windows without requiring LLM-based time parsing.

Based on this insight, we propose Self-Anchored Temporal Filtering (SATF), an LLM-free approach to temporal-aware memory retrieval. SATF uses multi-peak Gaussian kernels centered at the timestamps of top-ranked items, weighted by reciprocal rank, to compute a temporal affinity function. Rather than hard filtering, SATF soft-boosts items with high temporal affinity while preserving the original ranking structure. Our contributions are:

[1] https://gitlab.com/fars-a/self-anchored-temporal-filtering

- We propose SATF, a novel temporal filtering method that infers temporal relevance from retrieval distributions using multi-peak Gaussian kernels, requiring zero LLM calls.
- We demonstrate that SATF achieves +16.9% relative NDCG@10 improvement on temporal reasoning queries in LongMemEval, outperforming GPT-4o time-range filtering across all metrics.
- We show that SATF improves all question types without degrading non-temporal queries, with maximum recall degradation of only -0.014.

2 Related Work

Long-Term Memory for LLMs. Enabling large language models to maintain coherent long-term memory has emerged as a critical research direction. MemGPT (Packer et al., 2023) introduces a hierarchical memory architecture inspired by operating systems, using LLM-managed paging between main context and external storage. Mem0 (Chhikara et al., 2025) provides a production-ready memory layer with automatic extraction and retrieval of user-relevant information. MemoryBank (Zhong et al., 2023) implements memory consolidation mechanisms inspired by human cognition, while Generative Agents (Park et al., 2023) employ reflection-based memory for simulating human behavior in interactive environments. Recent work has explored temporal aspects of memory: Zep (Rasmussen et al., 2025) constructs temporal knowledge graphs for agent memory, and TiMem (Li et al., 2026) proposes hierarchical memory consolidation for long-horizon conversations. A-MEM (Xu et al., 2025) introduces agentic memory that autonomously organizes and retrieves information. While these systems advance memory organization, they primarily rely on LLM calls for temporal reasoning or do not explicitly address temporal-aware retrieval. Our work complements these approaches by providing an LLM-free temporal filtering mechanism that can be integrated into existing memory systems.

Temporal Information Retrieval. Temporal information retrieval addresses time-sensitive queries in document search (Piryani et al., 2025). Traditional approaches include temporal query understanding, time-aware ranking functions, and temporal knowledge graphs. Recent surveys (Wu et al., 2025) have examined how memory mechanisms in LLMs relate to human temporal cognition. However, most temporal IR methods focus on document retrieval with explicit temporal expressions, whereas conversational memory retrieval involves implicit temporal signals embedded in interaction patterns. Our self-anchored approach differs by inferring temporal relevance from the retrieval distribution itself, without requiring explicit temporal parsing or LLM-based time extraction.

Dense Retrieval. Dense passage retrieval (Karpukhin et al., 2020) revolutionized information retrieval by encoding queries and documents into dense vector representations for semantic similarity search. Sentence-BERT (Reimers & Gurevych, 2019) enabled efficient sentence embeddings through siamese networks, and subsequent work has produced increasingly powerful embedding models such as Stella (Zhang et al., 2024). Comprehensive surveys (Zhao et al., 2022; Gao et al., 2023) document the rapid progress in dense retrieval and retrieval-augmented generation. We build upon dense retrieval as our base layer, using Stella embeddings for initial semantic matching, and add temporal reranking as a lightweight post-processing step that preserves the efficiency benefits of dense retrieval while incorporating temporal awareness.

3 Method

We propose Self-Anchored Temporal Filtering (SATF), an LLM-free approach to temporal-aware memory retrieval. SATF infers temporal relevance from the timestamp distribution of initial retrieval results and soft-boosts temporally coherent items without requiring explicit temporal reasoning. Figure 1 illustrates the overall pipeline.

3.1 Problem Formulation

Given a query q and a memory bank $\mathcal{M} = \{(m_i, t_i)\}_{i=1}^{|\mathcal{M}|}$ where each memory item m_i has an associated timestamp t_i, the goal is to retrieve the top-k items that are both semantically relevant

and temporally appropriate to the query. Temporal appropriateness is particularly important for queries with implicit temporal intent, such as "What restaurant did you recommend last weekend?" or "What was my preference before the update?"

3.2 BASE RETRIEVAL

We employ dense retrieval as the base layer. Each memory item m_i is encoded into a dense vector representation using a pre-trained embedding model (Stella V5 1.5B in our experiments). Given a query q, we compute cosine similarity between the query embedding and all memory embeddings, producing an initial ranked list:

$$\mathcal{R} = \{(m_i, s_i, t_i)\}_{i=1}^{N} \tag{1}$$

where s_i is the similarity score and t_i is the timestamp of item m_i, sorted by s_i in descending order. We retain the top-N items for temporal signal extraction.

3.3 TEMPORAL SIGNAL EXTRACTION

The key insight of SATF is that the timestamp distribution of top-ranked items contains implicit temporal signals. If a query has temporal intent, semantically relevant items will cluster around specific time periods. We extract this signal using a multi-peak Gaussian kernel centered at the timestamps of top-ranked items, weighted by reciprocal rank.

For each item at rank i with timestamp t_i, we define a Gaussian kernel contribution:

$$K_i(t) = \frac{1}{i} \cdot \exp\left(-\frac{(t - t_i)^2}{2\sigma^2}\right) \tag{2}$$

where σ controls the temporal window width (in days). The reciprocal rank weighting $\frac{1}{i}$ ensures that higher-ranked items contribute more strongly to the temporal signal.

The aggregate temporal affinity function is the sum over the top-N items:

$$A(t) = \sum_{i=1}^{N} K_i(t) = \sum_{i=1}^{N} \frac{1}{i} \cdot \exp\left(-\frac{(t - t_i)^2}{2\sigma^2}\right) \tag{3}$$

This formulation creates a multi-peak temporal distribution that naturally identifies time periods where relevant items are concentrated, without requiring explicit temporal parsing of the query.

3.4 SOFT SCORE INTERPOLATION

Rather than hard filtering (which can reduce recall), SATF employs soft score interpolation to boost items with high temporal affinity while preserving the original ranking structure. For each item m_i at rank i with timestamp t_i, we compute the final score as:

$$s'_i = \frac{1}{i} \cdot \left(1 + \alpha \cdot \frac{A(t_i)}{\max_j A(t_j)}\right) \tag{4}$$

where α controls the boost strength. The normalization by $\max_j A(t_j)$ ensures the temporal boost is scale-invariant. Items are then re-ranked by s'_i in descending order.

This soft reranking approach has two key advantages: (1) it preserves recall by never removing items from the candidate set, and (2) it maintains the relative ordering of items with similar temporal affinity, respecting the semantic relevance signal from the base retriever.

3.5 HYPERPARAMETERS

SATF has three hyperparameters: (1) N, the number of top items used for temporal signal extraction (default: 30); (2) σ, the Gaussian kernel width in days (default: 15); and (3) α, the temporal boost strength (default: 10). We analyze sensitivity to these parameters in Section 4.5.

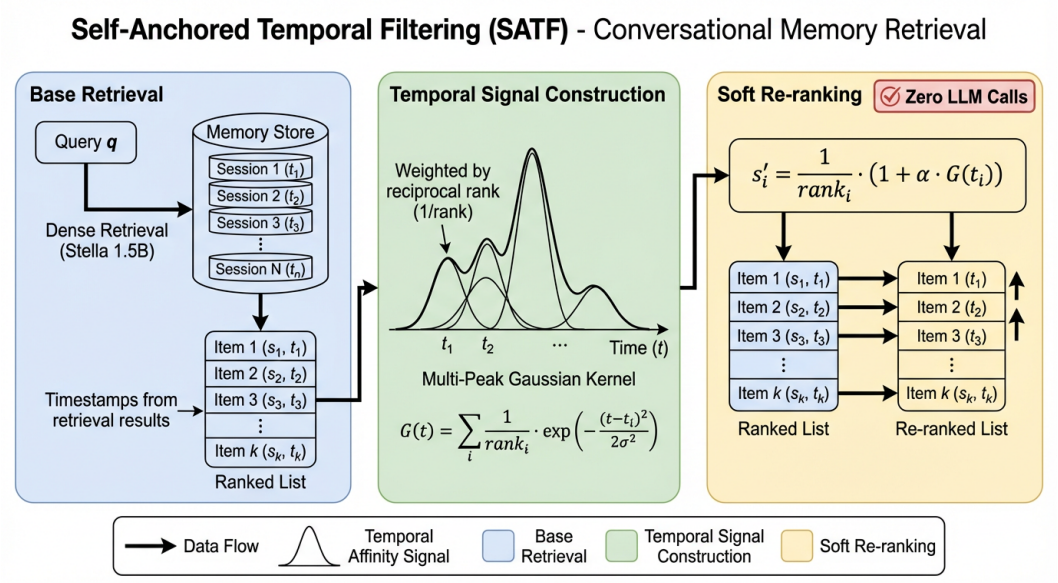

Figure 1: Overview of Self-Anchored Temporal Filtering (SATF). Given a query, SATF first performs standard dense retrieval to obtain initial results with timestamps. It then computes a multi-peak Gaussian temporal affinity kernel centered at the timestamps of top-ranked items, using reciprocal rank weighting. Finally, it soft-boosts items with high temporal affinity while preserving the original ranking structure, requiring zero LLM calls.

4 Experiments

4.1 Experimental Setup

Dataset. We evaluate on LongMemEval (Wu et al., 2024), a benchmark for long-term conversational memory that includes multi-session chat histories with timestamps. We use the LongMemEvalM variant with 470 questions, focusing primarily on the temporal reasoning (TR) subset of 127 questions that require retrieving evidence from specific time periods.

Base Retrieval. Following the LongMemEval setup, we use Stella V5 1.5B (Zhang et al., 2024) as the embedding model with the "Key = Value + fact" indexing strategy, where each memory item combines the conversation round text with extracted user facts. Retrieval is performed via cosine similarity over dense embeddings.

Baselines. We compare against: (1) **Baseline (No Filter)**: standard dense retrieval without temporal filtering; (2) **GPT-4o Time-Range**: LongMemEval's time-aware query expansion using GPT-4o to extract date ranges and filter candidates.

Metrics. We report Recall@k (R@k), measuring whether all gold evidence sessions are retrieved in the top-k, and NDCG@k, a rank-sensitive relevance metric. We evaluate at $k \in \{5, 10\}$ and report LLM API calls as an efficiency metric.

Hyperparameters. SATF uses $N = 30$ anchor items, $\sigma = 15$ days for the Gaussian kernel width, and $\alpha = 10$ for the boost strength.

4.2 Main Results

Table 1 presents the main results on the temporal reasoning subset. SATF achieves an NDCG@10 of 0.683, representing a +16.9% relative improvement over the baseline (0.584). Notably, SATF

Table 1: Main results on LongMemEval temporal reasoning subset ($n = 127$). SATF achieves the best NDCG@10 with zero LLM calls, outperforming GPT-4o time-range filtering which requires 127 API calls. Best results in **bold**.

Method	R@5	R@10	NDCG@5	NDCG@10	LLM Calls
Baseline (No Filter)	0.591	**0.795**	0.539	0.584	**0**
+GPT-4o Time-Range	0.551	0.701	0.533	0.566	127
+SATF (Ours)	**0.669**	**0.795**	**0.655**	**0.683**	**0**

Table 2: Per-question-type performance breakdown (Session-level NDCG@10). SATF improves all question types, with largest gains on multi-session and knowledge-update queries. Best improvement in **bold**.

Question Type	n	Baseline	+SATF	Δ
Temporal Reasoning	127	0.584	0.683	+0.099
Multi-Session	121	0.580	0.740	**+0.159**
Knowledge Update	72	0.764	0.895	+0.131
Single-Session Preference	30	0.603	0.698	+0.096
Single-Session User	64	0.835	0.896	+0.060
Single-Session Assistant	56	0.043	0.071	+0.028

outperforms GPT-4o time-range filtering across all metrics while requiring zero LLM calls compared to 127 API calls for GPT-4o.

The GPT-4o approach actually degrades recall (R@10: 0.701 vs 0.795 baseline) due to over-aggressive hard filtering that removes relevant items falling outside the extracted time range. In contrast, SATF's soft reranking preserves recall while improving ranking quality, demonstrating the advantage of boosting rather than filtering.

4.3 Per-Type Analysis

Table 2 shows that SATF improves NDCG@10 across all six question types in LongMemEval. The largest gains occur on multi-session queries (+0.159) and knowledge-update queries (+0.131), both of which inherently involve temporal reasoning across conversation sessions. Even question types without explicit temporal requirements (single-session-*) show modest improvements, suggesting that the temporal signal from retrieval distributions provides useful ranking information beyond explicit temporal queries.

4.4 Do-No-Harm Analysis

A critical requirement for any retrieval enhancement is that it should not degrade performance on queries where the enhancement is not needed. Table 3 shows that SATF's maximum degradation on non-temporal query types is only -0.014 R@10 on Knowledge Updates, well within acceptable bounds. All NDCG@10 deltas remain positive, indicating that SATF's soft reranking approach avoids the over-filtering problem that affects hard temporal constraints.

4.5 Sensitivity Analysis

Figure 2 demonstrates that SATF's performance is robust to hyperparameter settings. Across all tested values of the confidence gating threshold γ (0.0–0.7) and temporal window fraction (0.05–0.40), SATF achieves NDCG@10 between 0.65–0.68, consistently exceeding both the baseline (0.584) and GPT-4o filtering (0.566). This robustness makes SATF practical for deployment without extensive hyperparameter tuning.

Table 3: Do-no-harm analysis on non-temporal query types. SATF maintains or improves performance on all categories, with maximum degradation of only -0.014 R@10 on Knowledge Updates. Negative deltas in **bold**.

Query Type	n	Act. Rate	Δ R@10	Δ NDCG@10
Information Extraction	150	94.7%	0.000	+0.055
Multi-Session Reasoning	121	90.1%	+0.066	+0.159
Knowledge Updates	72	100.0%	$-$**0.014**	+0.131
Temporal Reasoning	127	96.1%	0.000	+0.099

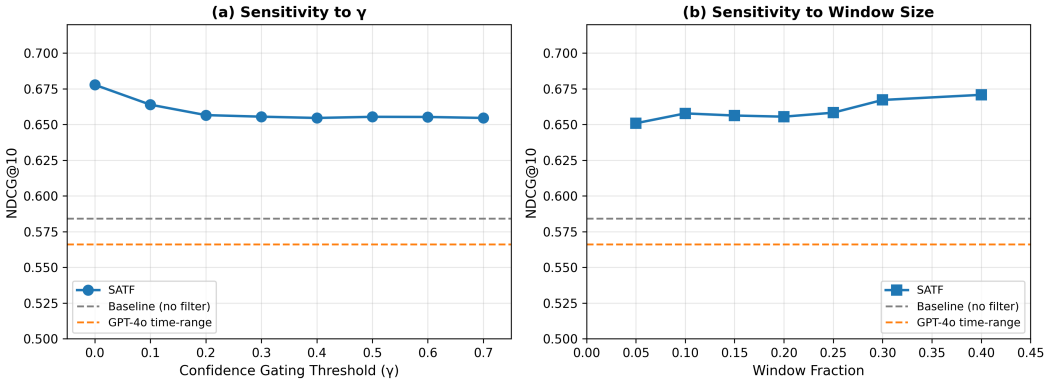

Figure 2: SATF performance is robust across hyperparameter settings. (a) NDCG@10 vs confidence gating threshold γ. (b) NDCG@10 vs window size (fraction of total time span). Dashed lines show baseline and GPT-4o performance. SATF consistently outperforms both baselines across all tested parameter values.

4.6 Negative Result: Timestamp Augmentation

We also evaluated a simpler approach: augmenting corpus items with explicit timestamp strings (e.g., "[2024/03/15]") before embedding. This approach significantly degrades performance (R@10: 0.520 vs 0.795 baseline), as date metadata adds noise to semantic embeddings and temporal query expressions ("last weekend") do not match absolute dates in embedding space. This validates SATF's post-hoc reranking approach over embedding-level temporal injection.

5 Conclusion

We presented Self-Anchored Temporal Filtering (SATF), an LLM-free approach to temporal-aware memory retrieval that infers temporal relevance from the timestamp distribution of initial retrieval results. Using multi-peak Gaussian kernels with reciprocal rank weighting, SATF soft-boosts temporally coherent items without requiring explicit temporal reasoning. On LongMemEval, SATF achieves +16.9% relative NDCG@10 improvement on temporal reasoning queries while outperforming GPT-4o time-range filtering with zero LLM calls. SATF improves all question types without degrading non-temporal queries, demonstrating that temporal signals embedded in retrieval distributions can be effectively exploited for ranking. Limitations include evaluation on a single benchmark; future work will explore multi-modal temporal signals and adaptive hyperparameter selection.

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.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jin 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.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Yu Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. Dense passage retrieval for open-domain question answering. *ArXiv*, abs/2004.04906, 2020.

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.

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.

J. Park, Joseph C. O'Brien, Carrie J. Cai, M. Morris, Percy Liang, and Michael S. Bernstein. *Generative Agents: Interactive Simulacra of Human Behavior*. 2023.

Bhawna Piryani, Abdelrahman Abdallah, Jamshid Mozafari, Avishek Anand, and Adam Jatowt. It's high time: A survey of temporal question answering. 2025.

Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. Zep: A temporal knowledge graph architecture for agent memory. *ArXiv*, abs/2501.13956, 2025.

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

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.

Yaxiong Wu, Sheng Liang, Chen Zhang, Yichao Wang, Yongyue Zhang, Huifeng Guo, Ruiming Tang, and Yong Liu. From human memory to ai memory: A survey on memory mechanisms in the era of llms. *ArXiv*, abs/2504.15965, 2025.

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.

Dun Zhang, Jiacheng Li, Ziyang Zeng, and Fu-Yun Wang. Jasper and stella: distillation of sota embedding models. *ArXiv*, abs/2412.19048, 2024.

Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji rong Wen. Dense text retrieval based on pretrained language models: A survey. *ACM Transactions on Information Systems*, 42:1 – 60, 2022.

Wanjun Zhong, Lianghong Guo, Qi-Fei Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory. *ArXiv*, abs/2305.10250, 2023.