MEMTRACE: FINE GRAINED SOURCE EVIDENCE RETRIEVAL FOR LONG-TERM CONVERSATIONAL MEMORY

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-term memory enables LLM-based agents to reuse information from past interactions. One approach to organizing these memories is to connect information across interactions in a graph. Subgraph retrieval then locates graph regions relevant to the current query. However, retrieving the source memories needed for answering still presents two challenges. First, retrieval guided by the overall semantics of a complex query may produce subgraphs that omit regions containing evidence relevant to only part of the question. Second, even when the required memories are retrieved, semantic similarity or graph importance may favor related memories that lack the required evidence, leaving supporting memories ranked lower. To address these challenges, we introduce MemTrace, a fine-grained source memory retrieval framework. MemTrace uses adaptive auxiliary queries alongside the original query to retrieve complementary local subgraphs. It traces retrieved relations to their source memories and combines aggregated relation support with contextual semantic similarity to rank candidates. Experiments on LongMemEval-S and LoCoMo show improvements in memory retrieval and downstream question answering. Controlled experiments confirm improved coverage of distributed evidence and more effective source ranking within a fixed candidate pool.

1 INTRODUCTION

LLM-based conversational agents need to draw on past interactions to maintain continuity across sessions and provide personalized responses (Pan et al., 2025; Kang et al., 2025). As interaction histories grow, retaining and effectively using this information within a limited context window becomes increasingly difficult (Liu et al., 2024; Packer et al., 2023; Du et al., 2025). Long-term memory systems address this challenge by storing and organizing information from past interactions for later retrieval, allowing agents to access historical information beyond the current context (Xu et al., 2025; Fang et al., 2026).

Memory systems organize past interactions into retrievable units, such as conversation turns, segments, or summaries (Pan et al., 2025; Xu et al., 2026). While these units preserve information from different portions of the history, connections between them often remain implicit. Graph-based approaches have emerged as a prominent direction for organizing and retrieving long-term memories, making these connections explicit by linking memory units or the entities and events they describe (Zhang et al., 2025; Tang et al., 2026; Jiang et al., 2026; Van et al., 2026). Their key advantage is the ability to combine query relevance with explicit connections among memories, helping retrieve evidence distributed across multiple interactions. In subgraph-based retrieval, systems identify relevant graph regions and select associated source memories as context for answering. However, locating relevant graph structure does not by itself determine which original memories supply the required evidence.

Two gaps remain in this retrieval process.

(1) Evidence discovery gap. Whole-query similarity may overlook graph regions containing important memories that match only a specific part of a complex question. Under a limited retrieval budget, these memories may never enter the candidate pool, as illustrated in Figure 1(a).

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.

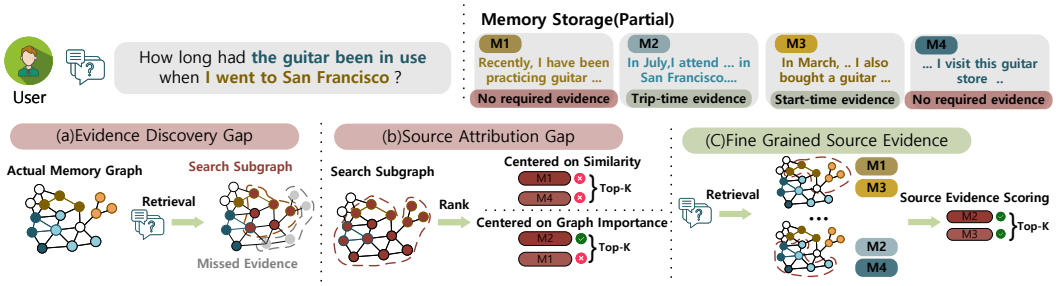

Figure 1: **Evidence discovery and source attribution in graph-based conversational memory retrieval.** The question requires complementary evidence from M2 and M3. **(a)** Retrieval guided by the full query may omit graph regions containing necessary evidence. **(b)** Even when the evidence is available, whole-query similarity can favor topical overlap, while graph importance does not distinguish the specific evidence supplied by associated source memories. Both can exclude required memories from the Top-K results. **(c)** MemTrace uses fine-grained information needs to guide local subgraph retrieval and traces local relational structure to source memories for evidence scoring, improving evidence coverage and source selection. Rankings are illustrative.

(2) Source attribution gap. Even when the required memories are retrieved, semantic similarity and graph importance may not distinguish their evidential contributions. A memory can closely match the query or be associated with prominent graph elements without stating the required fact, causing insufficient memories to rank above necessary evidence, as shown in Figure 1(b).

Existing graph-based memory systems have explored how to organize memories and locate relevant information through graph search and relevance propagation (Gutiérrez et al., 2024; Zhang et al., 2025; Van et al., 2026). These advances do not, however, fully resolve how the retrieved graph structure should inform the selection of source evidence. When source memories are ranked by semantic similarity or scores projected from the graph, their priority reflects textual relevance or the importance of associated graph elements. The local relations within a retrieved subgraph provide an additional basis for source assessment: their provenance records which original memories support the retrieved relations. Our work focuses on using these correspondences to assess the contributions of individual source memories, rather than treating source selection solely as a consequence of semantic or graph-level relevance.

To address this problem, we propose MemTrace, a fine-grained source evidence retrieval framework that uses local relational structure to guide source memory ranking. MemTrace comprises two key modules: **(1) Evidence Discovery:** it adaptively identifies information needs requiring separate lookups and guides an independent local subgraph search for each, providing candidate memories and local relations for the different needs. **(2) Source Attribution:** it traces each subgraph's local relational structure back to its source memories and aggregates relation provenance support for each candidate. This support is combined with contextual semantic similarity to rank memories for answering. Need-guided retrieval thus provides the local graph structure used by our source evidence scoring mechanism.

In summary, our main contributions are as follows.

- We characterize the source attribution gap in graph-based memory retrieval: relevance to a query or graph region does not necessarily reflect the evidence supplied by a source memory.
- We propose MemTrace, which retrieves local subgraphs through adaptive queries and traces their relations to source memories, combining relation support with semantic relevance for source evidence ranking.
- Experiments on LongMemEval-S and LoCoMo show improved memory retrieval and downstream QA. Controlled experiments further demonstrate the effectiveness of source evidence scoring in prioritizing supporting memories.

2 Related Work

Long-Term Memory for Conversational Agents. Existing research explores how to construct, organize, and maintain memories from interaction histories. For memory construction, SeCom partitions conversations into semantically coherent segments, while LightMem emphasizes efficient memory construction and maintenance (Pan et al., 2025; Fang et al., 2026). Memory management is another focus: RMM uses reflection to manage stored memories, and MemoryOS coordinates memory across storage levels (Tan et al., 2025; Kang et al., 2025). Other approaches explicitly organize connections among memories. A-Mem builds linked memory representations, MemGAS connects information across multiple granularities, MAGMA organizes memory through multiple graph views, and Mnemis adopts a hierarchical graph structure (Xu et al., 2025; 2026; Jiang et al., 2026; Tang et al., 2026). These systems provide different representations and management mechanisms for retaining and accessing historical information. Building on structured memory, our work focuses on using retrieved local relational structure to assess the evidential contributions of source memories.

Memory Retrieval and Evidence Discovery. Existing methods improve evidence discovery through different retrieval granularities and search strategies. SeCom retrieves semantically coherent conversation segments, MemGAS explores multi-granularity retrieval, and Mnemis combines similarity search with hierarchical selection (Pan et al., 2025; Xu et al., 2026; Tang et al., 2026). Reasoning and reflection also inform memory retrieval: Associa incorporates deliberate recall, MemR^3 uses reflection and evidence-gap tracking, and MRAgent reconstructs memory through reasoning-guided graph traversal (Zhang et al., 2025; Du et al., 2025; Ji et al., 2026). MemTrace uses adaptive auxiliary queries to discover complementary evidence through independent local subgraph searches, providing candidate memories and relational structure for subsequent fine-grained source evidence assessment.

Source Attribution and Evidence Ranking. Beyond evidence discovery, existing work explores graph-based ranking and provenance mechanisms for selecting source evidence. In graph-based document retrieval, HippoRAG aggregates Personalized PageRank (PPR) scores from entities to passages, while HippoRAG 2 integrates passage nodes into the graph and ranks them through PPR (Gutiérrez et al., 2024; 2025). For conversational memory, MemGAS initializes PPR with multi-granularity relevance scores and filters retrieved memories using an LLM (Xu et al., 2026). Semantic signals also inform graph propagation: EHRAG combines structural and semantic diffusion with PPR refinement, while MemORAI uses query-conditioned edge weights to rank source turns through Dynamic Weighted PageRank (Song et al., 2026; Van et al., 2026). Provenance supports evidence recovery as well: MemORAI returns triples associated with selected turns, and SEEM expands retrieved event frames into their supporting source passages (Van et al., 2026; Lu et al., 2026). MemTrace uses local relation provenance directly in source scoring: it traces retrieved subgraph relations to candidate memories and aggregates their query-relevant support, combining this signal with contextual semantic similarity to rank source evidence.

3 Method

Whole-query retrieval may miss evidence across graph regions, while ranking may fail to distinguish source memories' evidential contributions. MemTrace addresses these problems through evidence discovery and source attribution (Figure 2): adaptive queries retrieve complementary local subgraphs, whose relation provenance supports source scoring alongside semantic similarity. We first define the task, then describe query decomposition and local retrieval, and finally present source scoring and selection.

3.1 Problem Formulation

Given a conversation history $\mathcal{H} = \{u_t\}_{t=1}^{T}$ organized into sessions and a query Q, the task is to return a ranked list of at most K distinct sessions containing the evidence needed to answer Q. We assume a typed memory graph $G = (V, E)$ constructed from $\mathcal{H}$, with provenance mappings $\mathrm{Prov}(v)$ and $\mathrm{Prov}(e)$ from nodes and relations to their source turns. Retrieval first identifies relevant subgraphs. Their provenance determines the candidate turns, whose scores are aggregated to rank sessions for the downstream answer model.

3.2 Adaptive Query Decomposition and Entry Typing

A complex query may require evidence from different regions of the memory graph. Whole-query retrieval can overlook regions relevant to only part of the question. We therefore use an LLM to generate auxiliary queries for information needs requiring separate memory lookups and assign each query an entry type from the predefined routing type set $\mathcal{A}$:

$$\mathcal{D}(Q) = \{(q_i, a_i)\}_{i=1}^{M(Q)}, \qquad a_i \in \mathcal{A}. \tag{1}$$

The entry type focuses initial node selection on the category of information sought by the auxiliary query, while subsequent graph search can expand across types. The graph construction prompt is given in Appendix A.1; the routing type definitions and examples are provided in Appendix A.3.

We generate auxiliary queries only when answering requires separate memory lookups. Each auxiliary query targets an information need rather than an individual memory, allowing local graph search to gather supporting memories through their connections. When separate lookups are unnecessary, the auxiliary set is empty. The original query is retained alongside any generated auxiliary queries:

$$\mathcal{Q}(Q) = \{Q\} \cup \{q_i\}_{i=1}^{M(Q)}. \tag{2}$$

Each query guides an independent local subgraph search, giving less prominent information needs separate retrieval opportunities while preserving the full question as a retrieval view. The full decomposition prompt and operational rules are provided in Appendix A.2.

3.3 Independent Local Subgraph Retrieval

We adopt the PCST-based subgraph retrieval formulation of G-Retriever (He et al., 2024), augmented with the type-based entry selection described in Section 3.2. We independently retrieve a local subgraph for the original query and each auxiliary query, then merge and deduplicate their associated source turns into a shared candidate pool. We retain each query's selected relations, relation prizes, and provenance mappings for source evidence scoring in Section 3.4. This preserves query-specific relational support for assessing candidate memories. Prize assignment, the PCST objective, and retrieval configurations are provided in Appendix D.

3.4 Source Memory Ranking with Relation Provenance

Retrieving a relevant subgraph does not determine which associated source memories should be prioritized. Memories linked to the same subgraph may support different subsets of its relations, a distinction that semantic similarity or graph importance alone may not capture. We therefore use relation provenance to connect the retrieved relational structure to individual source memories and aggregate their support for ranking.

3.4.1 Source Evidence Scoring

For retrieval query q_i, let $G_i^* = (V_i^*, E_i^*)$ denote its retrieved subgraph, with $q_0 = Q$. We score every turn in the shared candidate pool under each query.

Relation provenance support. Let $\mathrm{Prov}(e)$ record the source turns supporting relation e, and let $p_i(e)$ be its query-dependent prize assigned during retrieval. For candidate turn u, we aggregate the prizes of the selected relations attributed to it:

$$g_i(u) = \sum_{e \in E_i^*} p_i(e)\, \mathbb{I}[u \in \mathrm{Prov}(e)]. \tag{3}$$

Although candidates may originate from the same retrieved subgraph, this aggregation distinguishes them according to which query-relevant relations they support. A memory receives contributions only from relations that trace back to it, with larger contributions from higher-prize relations. A candidate without provenance links to the selected relations receives zero support. Thus, the retrieved subgraph supplies both candidate memories and a relational basis for comparing them.

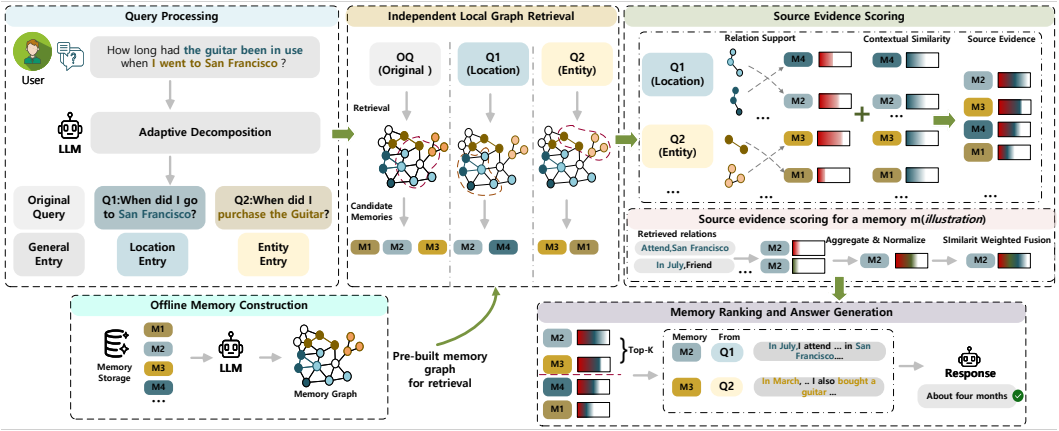

Figure 2: **Overview of MemTrace.** The original query and adaptively generated auxiliary queries guide independent local subgraph retrieval, with auxiliary queries using type-based entry selection. Retrieved relations are traced to their source memories, and their query-dependent prizes are aggregated and normalized to obtain relation support. This support is combined with contextual semantic similarity to score candidate memories. Scores are aggregated across queries for evidence selection and answer generation; session-level retrieval further aggregates turn scores within each session. The inset illustrates relation support aggregation for a candidate memory.

We normalize the accumulated support by the maximum value among the source turns $\mathcal{U}_i$ associated with the same subgraph:

$$s_{\text{prov}}(i,u) = \begin{cases} \dfrac{g_i(u)}{\max_{u' \in \mathcal{U}_i} g_i(u')}, & \max_{u' \in \mathcal{U}_i} g_i(u') > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

This places provenance support in $[0,1]$ for combination with semantic similarity. The inset in Figure 2 illustrates how retrieved relations contribute to a candidate memory's support score.

Contextual semantic relevance. Graph extraction may not preserve all details needed to assess a source memory. We therefore complement relation support with semantic similarity computed from the original text and its context. For candidate turn u_t, we concatenate the turn and its two neighboring turns on either side:

$$C_{\pm 2}(u_t) = [u_{t-2}; u_{t-1}; u_t; u_{t+1}; u_{t+2}], \tag{5}$$

and compute

$$s_{\text{sem}}(i,u_t) = \max(0, \cos(\mathbf{e}(q_i), \mathbf{e}(C_{\pm 2}(u_t)))) \,. \tag{6}$$

The window provides context for semantic matching; provenance support remains assigned to the center turn's identifier.

We combine the two signals into a source evidence score:

$$s_i(u) = \alpha \, s_{\text{sem}}(i,u) + (1-\alpha) \, s_{\text{prov}}(i,u), \tag{7}$$

where $\alpha = 0.8$ in the main configuration. The combined score rewards both contextual relevance and support for locally retrieved relations, while allowing textual relevance to complement incomplete graph representations. It serves as a ranking signal rather than an entailment probability. Weight sensitivity is examined in Appendix C.4.

3.4.2 Score Aggregation and Final Selection

A memory may supply evidence for only one part of a complex question. To preserve such complementary contributions, we assign each candidate its highest score across the original and auxiliary queries:

$$S(u) = \max_{i \in \{0,\ldots,M(Q)\}} s_i(u). \tag{8}$$

This allows strong support for a single information need to determine a memory's priority without requiring it to match every query.

For the session-level task defined in Section 3.1, turn scores serve as intermediate evidence scores. Each candidate session receives the highest score among its candidate turns:

$$S_{\text{session}}(s) = \max_{u:\text{Session}(u)=s} S(u). \tag{9}$$

This lets a strongly supported turn determine its session's priority without accumulating an advantage merely from having more candidate turns. We rank candidate sessions by these scores and select the Top-K with deterministic tie breaking and no fixed quota for individual queries. The selected evidence units are provided to the answer model under the corresponding retrieval protocol. The turn-level selection variant is specified in Appendix A.4.

This design supports complementary evidence discovery and uses local relational support to distinguish candidate memories, connecting graph retrieval to source evidence selection.

4 Experiments

Our experiments ask whether MemTrace improves retrieval and downstream QA, how decomposition compares with widening a single graph search, and whether relation provenance improves session-level retrieval beyond semantic similarity or graph centrality when the source-turn candidate pool is fixed.

4.1 Experimental Setup

We follow the experimental setup of MemGAS (Xu et al., 2026) in our baseline selection, downstream QA metrics, and choice of answer generator and judge. Dataset-specific protocols and MemTrace configurations are detailed below.

Datasets and Evaluation Metrics. We evaluate on **LongMemEval-S** (Wu et al., 2025) and **LoCoMo** (Maharana et al., 2024). For LongMemEval-S, we follow the evaluation settings of MemGAS (Xu et al., 2026). The generation and evaluation prompts for LongMemEval-S and LoCoMo are provided in Appendices A.4.1 and A.4.2, respectively. Retrieval performance is reported using Recall@K (R@K) and NDCG@K. For both benchmarks, R@K denotes Recall-all: a query is counted as correct at K only when all Gold Sessions are included in the Top-K retrieved results.

To comprehensively evaluate downstream QA performance, we report multiple complementary metrics. Following Maharana et al. (2024), we use token-level F1, together with BLEU-4, BERTScore, and ROUGE-1/2/L following Pan et al. (2025). We additionally report GPT-4o-as-Judge (4o-J), following the LLM-as-a-Judge evaluation paradigm of Zheng et al. (2023), where GPT-4o assesses whether the generated response is consistent with and correctly answers the question according to the reference answer.

Baselines. For retrieval, we compare with the dense retriever Contriever (Izacard et al., 2022), SeCom, which organizes memory into semantic segments (Pan et al., 2025), and the graph retrievers HippoRAG 2 (Gutiérrez et al., 2025) and MemGAS (Xu et al., 2026). For downstream QA, we additionally report Full History, MPNet, MPC, and RecurSum, together with RAPTOR (Sarthi et al., 2024) and A-Mem (Xu et al., 2025). These baseline results are taken from MemGAS (Xu et al., 2026).

Implementation Details. The intact query is always retained together with the auxiliary subqueries generated by the decomposer. Each retrieval view independently runs PCST, following its use for compact graph retrieval conditioned on the query in G-Retriever (He et al., 2024), with Node-K = Edge-K = 15 and a fixed edge cost of 0.5.

Source evidence scoring uses $\alpha = 0.8$, and candidates from all retrieval views are merged and ranked by the final source evidence score. For downstream QA, the Top-3 retrieved sessions are provided to GPT-4o-mini and GPT-4o is used as the judge. Prompt details, routing attributes, and additional implementation settings are given in the appendix. The appendix further shows that retrieval performance is stable across the evaluated decomposition models.

Table 1: Session-level retrieval performance (%). LongMemEval-S contains 470 official questions that do not require refusal; LoCoMo contains 1,531 Category 1–4 questions with valid Gold Session mappings. Bold and underlined values denote the best and second-best result in each dataset block, respectively.

	Top-3	Top-5		Top-10	
Method	R@3	R@5	NDCG@5	R@10	NDCG@10
LongMemEval-S					
Contriever	71.06	81.28	82.47	90.00	84.29
HippoRAG 2	75.53	84.68	87.32	91.28	88.73
MemGAS	<u>78.51</u>	<u>88.94</u>	<u>88.77</u>	<u>94.47</u>	<u>89.96</u>
MemTrace	**85.53**	**94.04**	**95.28**	**97.45**	**95.75**
LoCoMo					
Contriever	49.90	58.26	56.29	71.80	60.92
SeCom	55.24	64.80	62.36	78.30	66.97
HippoRAG 2	56.60	65.06	62.50	78.05	66.79
MemGAS	<u>57.30</u>	<u>67.32</u>	<u>63.62</u>	<u>81.82</u>	<u>68.42</u>
MemTrace	**67.28**	**74.53**	**69.67**	**83.74**	**72.79**

4.2 Overall Results

Memory Retrieval. The main retrieval experiment (Table 1) reports Session-level retrieval, measuring whether the required memories are retained under a fixed Top-K budget.

MemTrace obtains the highest reported recall and NDCG in both benchmark blocks. Because these rows come from independently implemented systems, the main retrieval comparison establishes overall competitiveness rather than attributing the gain to a particular module. We isolate evidence discovery and source attribution in the component ablation and controlled experiments below.

Downstream Question Answering. The downstream QA experiment (Table 2) reports performance using the same retrieved evidence. MemTrace directly answers from the retrieved Top-3 sessions, while MemTrace* applies an optional LLM evidence filter, analogous to the evidence compression stage evaluated by MemGAS, to the same sessions before generation. The filter therefore changes only the reader input and does not modify retrieval.

4.3 Component Ablation

To isolate the contribution of each component, the ablation in Table 3 progressively adds adaptive query decomposition, typed routing, and source scoring on the same 470 LongMemEval-S questions. All variants use the same final candidate selection procedure.

Decomposition and typed routing improve R@10 by 4.04 percentage points, but do not improve R@3, suggesting that gains at larger retrieval cutoffs do not necessarily translate into better top-ranked results. Adding source evidence scoring further improves R@3 by 3.62 percentage points and NDCG@5 by 2.51 points. These results highlight the importance of distinguishing the evidential contributions of retrieved memories when only a few can be selected.

This complementary pattern motivates the controlled experiments below: decomposition is tested against a wider single route, and source scoring is tested under an identical candidate pool.

4.4 Core Component Control Experiments

4.4.1 Is Evidence Decomposition Only Expanding the Search Scope?

We examine whether widening a single retrieval route can reproduce the candidate coverage gains of adaptive query decomposition. Figure 3 compares query-only retrieval with node/edge preselection budgets ranging from 15/15 to 30/30 against decomposition with a budget of 15/15 per retrieval view. Graph construction, the PCST algorithm, provenance mapping, and source turn scoring remain

Method	4o-J	F1	B-4	R-1	R-2	R-L	BS	Avg. Tokens
LongMemEval-S								
Contriever	55.40	13.78	2.21	14.46	6.93	12.89	83.70	8,286
MPC	53.80	13.60	1.74	14.27	6.49	12.95	83.49	8,457
RecurSum	35.40	12.29	2.09	13.01	5.55	11.52	83.60	8,853
SeCom	56.00	12.95	2.25	13.80	6.09	11.93	83.51	2,741
HippoRAG 2	57.60	14.73	2.15	15.30	7.36	13.83	83.86	8,530
RAPTOR	32.20	12.08	1.90	12.73	5.82	11.25	83.50	6,254
A-Mem	55.60	13.73	2.11	14.82	6.81	12.98	83.88	9,018
MemGAS	60.20	<u>20.38</u>	<u>4.22</u>	<u>21.05</u>	<u>10.47</u>	<u>19.47</u>	<u>85.21</u>	8,829
MemTrace	**68.20**	16.36	2.38	16.97	8.44	15.62	84.10	8,777
MemTrace*	<u>66.40</u>	**22.42**	**5.33**	**23.07**	**11.83**	**21.60**	**85.66**	8,917
LoCoMo								
Full History	33.43	12.23	1.84	12.70	5.66	11.73	84.07	20,078
MPNet	38.07	14.44	2.35	14.90	6.83	13.90	84.42	2,472
Contriever	40.33	15.66	2.67	16.01	7.68	15.00	84.65	2,348
MPC	40.38	14.81	1.99	15.10	6.83	14.13	84.43	2,683
RecurSum	22.56	9.14	0.99	9.82	3.38	8.98	83.45	3,074
SeCom	44.21	13.79	2.30	14.28	6.17	13.30	84.04	1,021
HippoRAG 2	45.62	16.66	2.91	17.01	8.27	15.93	84.88	2,991
RAPTOR	31.72	14.55	2.88	15.09	7.49	14.18	84.48	1,931
A-Mem	40.81	14.72	2.83	16.22	7.71	14.89	84.72	3,042
MemGAS	41.07	17.66	3.61	18.00	8.93	16.99	85.13	2,825
MemTrace	<u>52.37</u>	<u>22.20</u>	<u>4.67</u>	<u>22.58</u>	<u>13.64</u>	<u>21.68</u>	<u>86.04</u>	3,963
MemTrace*	**52.47**	**26.92**	**6.61**	**27.29**	**16.92**	**26.35**	**86.80**	4,075

Table 2: Downstream QA performance (%). B-4 denotes BLEU-4, R-1/R-2/R-L denote ROUGE-1/2/L, and BS denotes BERTScore. MemTrace* applies an optional LLM evidence filter to the same retrieved Top-3 sessions. Bold and underlined values indicate the best and second-best result in each dataset block, respectively.

Table 3: Component ablation on LongMemEval-S, Session-level (%). D: decomposition; T: typed routing; P: PCST; S: source scoring.

Method	D	T	P	S	R@3	R@5	NDCG@5	R@10	NDCG@10	Time(s)
Original Query	×	×	✓	×	82.55	91.28	92.77	92.98	93.30	0.61
+ Decomposition	✓	×	✓	×	82.13	91.70	92.90	94.47	93.64	0.92
+ Typed Routing	✓	✓	✓	×	81.91	91.70	92.77	97.02	93.88	1.09
+ Source Scoring	✓	✓	✓	✓	**85.53**	**94.04**	**95.28**	**97.45**	**95.75**	1.10

unchanged. This comparison evaluates two ways of organizing retrieval without matching their computational cost.

Widening the original-query route improves candidate recall, but decomposition remains stronger when evidence spans multiple sessions. Compared with Q-only 30/30, Decomposition 15/15 improves candidate recall by 8.39 percentage points for questions with two Gold Sessions and 5.37 points for those with at least three. Decomposition selects fewer graph nodes and edges per question on average, while producing larger candidate pools. For questions with one Gold Session, however, Q-only 30/30 achieves 1.97 percentage points higher candidate recall. These results suggest that, compared with simply expanding the search scope, adaptive query decomposition retrieves the required evidence more effectively when it is distributed across multiple sessions.

4.4.2 Source Evidence Ranking with a Fixed Candidate Pool

Table 4 compares the Session-level retrieval performance of semantic similarity, graph ranking using PPR, and source evidence scoring on the same source-turn candidate pool. Multiview Semantic Similarity takes the maximum contextual similarity across the original query and all auxiliary subqueries. Following the use of PPR for graph memory retrieval in HippoRAG 2, MemGAS,

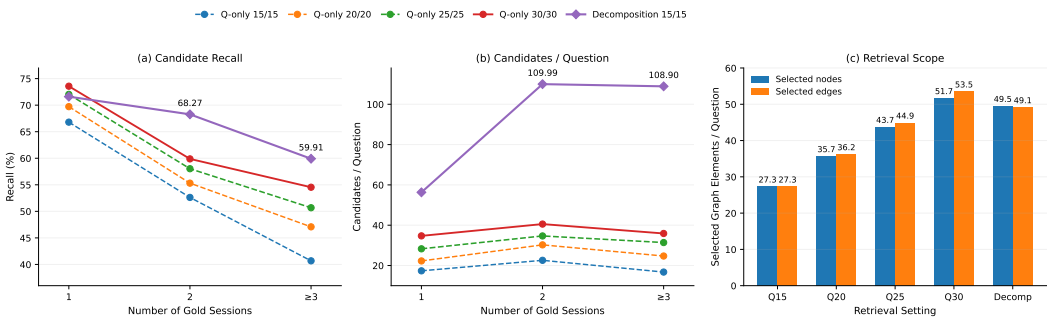

Figure 3: **Expansion of a single route versus adaptive query decomposition on LoCoMo.** (a) Candidate recall and (b) the deduplicated candidate pool per question, grouped by the number of Gold Sessions. Candidate recall measures the fraction of each question's annotated source turns present in the candidate pool before final ranking, averaged within each group. (c) The overall mean number of nodes and edges selected by PCST per question. Query-only widens a single retrieval route, whereas Decomposition 15/15 uses independent routes for the intact query and auxiliary subqueries.

Table 4: LoCoMo Session-level source evidence ranking (%) and throughput. $R@K$ denotes Recall-all. Queries/s is the number of questions processed per second during computation of ranking signals and final sorting; higher is faster. Accuracy uses 1,531 questions, while timing uses all 1,536 frozen records and excludes model loading, context embedding, candidate retrieval, and evaluation.

Method	R@3	R@5	NDCG@5	R@10	NDCG@10	Queries/s
Original Query Similarity	61.92	71.00	65.04	81.65	68.79	980.6
Multiview Semantic Similarity	61.66	70.80	64.52	81.58	68.37	841.4
Original Query PPR	59.96	68.13	61.51	80.47	65.77	142.6
PPR-RRF	58.46	67.93	60.55	81.19	64.99	81.7
Source Evidence Score	**67.28**	**74.53**	**69.67**	**83.74**	**72.79**	210.7

and Associa (Gutiérrez et al., 2025; Xu et al., 2026; Zhang et al., 2025), our PPR baselines run personalized PageRank within the frozen PCST subgraphs, using positive cosine similarity between the retrieval view and each node as the personalization seed, and project node scores to candidate turns through node provenance. Original Query PPR uses only the original query view. PPR-RRF ranks candidate turns separately under each retrieval view and combines these rankings using reciprocal rank fusion (RRF) (Cormack et al., 2009). Source Evidence Score follows Equations 7 and 8, without additional rank normalization. Throughput measures each signal's computation and final sorting from cached embeddings and frozen PCST outputs.

Source Evidence Score outperforms the semantic and PPR baselines on all reported retrieval metrics, supporting the value of relation provenance for source ranking. Appendix C.2 examines how these results vary with evidence distribution and whether allocating retrieval slots across views helps when evidence spans multiple sessions. Appendix B provides qualitative examples of evidence discovery and source attribution.

5 Conclusion

We introduced MemTrace, a retrieval framework for fine grained evidence that separates evidence discovery from source memory attribution. Subqueries guide independent graph searches for distinct information needs, while relation provenance links selected graph structure to supporting source memories for source evidence ranking. Across LongMemEval-S and LoCoMo, this design improves session-level retrieval and downstream QA. Controlled experiments show that decomposition improves candidate coverage for distributed evidence and that provenance-based source scoring improves session-level retrieval within a fixed candidate pool.

AI USE STATEMENT

During the preparation of this manuscript, the authors used AI-based tools for assistance with language refinement, organization of technical writing, and improving the clarity of explanations. All scientific ideas, experimental designs, implementations, results, analyses, and conclusions were developed and verified by the authors. The authors take full responsibility for the content of this manuscript.

REPRODUCIBILITY STATEMENT

The appendix documents the graph construction and decomposition prompts, routing attributes, PCST formulation, retrieval hyperparameters, QA evaluation protocols, and additional controlled analyses. We specify the components held fixed in each controlled comparison, including query sets, candidate pools, provenance mappings, and evaluators where applicable. We plan to publicly release the code and experimental configurations to facilitate reproduction of our results.

REFERENCES

Gordon V. Cormack, Charles L. A. Clarke, and Stefan Buettcher. Reciprocal Rank Fusion Outperforms Condorcet and Individual Rank Learning Methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 758–759, 2009. doi: 10.1145/1571941.1572114. URL https://doi.org/10.1145/1571941.1572114.

Xingbo Du, Loka Li, Duzhen Zhang, and Le Song. MemR3: Memory Retrieval via Reflective Reasoning for LLM Agents. *arXiv preprint arXiv:2512.20237*, 2025. doi: 10.48550/arXiv.2512.20237. URL https://arxiv.org/abs/2512.20237.

Jizhan Fang, Xinle Deng, Haoming Xu, Ziyan Jiang, Yuqi Tang, Ziwen Xu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, Huajun Chen, and Ningyu Zhang. LightMem: Lightweight and Efficient Memory-Augmented Generation. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/a05b72653ec5b473732129829ae04195-Abstract-Conference.html.

Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. HippoRAG: Neurobiologically Inspired Long-Term Memory for Large Language Models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 59532–59569, 2024. doi: 10.52202/079017-1902. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/6ddc001d07ca4f319af96a3024f6dbd1-Abstract.html.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From RAG to Memory: Non-Parametric Continual Learning for Large Language Models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 21497–21515, 2025. URL https://proceedings.mlr.press/v267/gutierrez25a.html.

Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V. Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. G-Retriever: Retrieval-Augmented Generation for Textual Graph Understanding and Question Answering. In *Advances in Neural Information Processing Systems*, volume 37, pp. 132876–132907, 2024. doi: 10.52202/079017-4224. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/efaf1c9726648c8ba363a5c927440529-Abstract-Conference.html.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised Dense Information Retrieval with Contrastive Learning. *Transactions on Machine Learning Research*, 2022. URL https://openreview.net/forum?id=jKN1pXi7b0.

Shuo Ji, Yibo Li, and Bryan Hooi. Memory is Reconstructed, Not Retrieved: Graph Memory for LLM Agents. *arXiv preprint arXiv:2606.06036*, 2026. doi: 10.48550/arXiv.2606.06036. URL https://arxiv.org/abs/2606.06036.

Dongming Jiang, Yi Li, Guanpeng Li, and Bingzhe Li. MAGMA: A Multi-Graph based Agentic Memory Architecture for AI Agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 36848–36865, 2026. doi: 10.18653/v1/2026.acl-long.1709. URL https://aclanthology.org/2026.acl-long.1709/.

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. Memory OS of AI Agent. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 25961–25970, 2025. doi: 10.18653/v1/2025.emnlp-main.1318. URL https://aclanthology.org/2025.emnlp-main.1318/.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the Middle: How Language Models Use Long Contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024. doi: 10.1162/tacl_a_00638. URL https://aclanthology.org/2024.tacl-1.9/.

Zhengxuan Lu, Dongfang Li, Yukun Shi, Beilun Wang, Longyue Wang, and Baotian Hu. Structured Episodic Event Memory. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6125–6141, 2026. doi: 10.18653/v1/2026.acl-long.277. URL https://aclanthology.org/2026.acl-long.277/.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating Very Long-Term Conversational Memory of LLM Agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13851–13870, 2024. doi: 10.18653/v1/2024.acl-long.747. URL https://aclanthology.org/2024.acl-long.747/.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. MemGPT: Towards LLMs as Operating Systems. *arXiv preprint arXiv:2310.08560*, 2023. doi: 10.48550/arXiv.2310.08560. URL https://arxiv.org/abs/2310.08560.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Xufang Luo, Hao Cheng, Dongsheng Li, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Jianfeng Gao. On Memory Construction and Retrieval for Personalized Conversational Agents. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/e56f394bbd4f0ec81393d767caa5a31b-Abstract-Conference.html.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. RAPTOR: Recursive Abstractive Processing for Tree-Organized Retrieval. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/8a2acd174940dbca361a6398a4f9df91-Abstract-Conference.html.

Yifan Song, Xingjian Tao, Zhicheng Yang, Yihong Luo, and Jing Tang. EHRAG: Bridging Semantic Gaps in Lightweight GraphRAG via Hybrid Hypergraph Construction and Retrieval. *arXiv preprint arXiv:2604.17458*, 2026. doi: 10.48550/arXiv.2604.17458. URL https://arxiv.org/abs/2604.17458.

Zhen Tan, Jun Yan, I-Hung Hsu, Rujun Han, Zifeng Wang, Long T. Le, Yiwen Song, Yanfei Chen, Hamid Palangi, George Lee, Anand Iyer, Tianlong Chen, Huan Liu, Chen-Yu Lee, and Tomas Pfister. In Prospect and Retrospect: Reflective Memory Management for Long-term Personalized Dialogue Agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8416–8439, 2025. doi: 10.18653/v1/2025.acl-long.413. URL https://aclanthology.org/2025.acl-long.413/.

Zihao Tang, Xin Yu, Ziyu Xiao, Zengxuan Wen, Zelin Li, Jiaxi Zhou, Hualei Wang, Haohua Wang, Haizhen Huang, Denvy Deng, Feng Sun, and Qi Zhang. Mnemis: Dual-Route Retrieval on Hierarchical Graphs for Long-Term LLM Memory. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23914–23928, 2026. doi: 10.18653/v1/2026.acl-long.1096. URL https://aclanthology.org/2026.acl-long.1096/.

Hung Pham Van, Nguyen Manh Hieu, Khang Pham Tran Tuan, Nam Le Hai, Linh Ngo Van, Diep Thi-Ngoc Nguyen, and Trung Le. MemORAI: Memory Organization and Retrieval via Adaptive Graph Intelligence for LLM Conversational Agents. In *Findings of the Association for Computational*

Linguistics: ACL 2026, pp. 28235–28253, 2026. doi: 10.18653/v1/2026.findings-acl.1408. URL https://aclanthology.org/2026.findings-acl.1408/.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. LongMemEval: Benchmarking Chat Assistants on Long-Term Interactive Memory. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.10813.

Derong Xu, Yi Wen, Pengyue Jia, Yingyi Zhang, Wenlin Zhang, Yichao Wang, Huifeng Guo, Ruiming Tang, Xiangyu Zhao, Enhong Chen, and Tong Xu. From Single to Multi-Granularity: Toward Long-Term Memory Association and Selection of Conversational Agents. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/93e444b4a1a64815631c3c74977ee200-Abstract-Conference.html.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-Mem: Agentic Memory for LLM Agents. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-0593. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/19909c36f51abc4856b4560aff3d36d6-Abstract-Conference.html.

Yujie Zhang, Weikang Yuan, and Zhuoren Jiang. Bridging Intuitive Associations and Deliberate Recall: Empowering LLM Personal Assistant with Graph-Structured Long-term Memory. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 17533–17547, 2025. doi: 10.18653/v1/2025.findings-acl.901. URL https://aclanthology.org/2025.findings-acl.901/.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems*, volume 36, pp. 46595–46623. Curran Associates, Inc., 2023. doi: 10.52202/075280-2020. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/91f18a1287b398d378ef22505bf41832-Paper-Datasets_and_Benchmarks.pdf.

APPENDIX TABLE OF CONTENTS

A Prompts and Implementation Details 13
A.1 Memory Graph Construction Prompt 13
A.2 Adaptive Query Decomposition Prompt 14
A.3 Graph Entry Types and Examples 17
A.4 Answer Generation and Evaluation Protocols 18
B Qualitative Case Studies 20
C Additional Analyses 21
C.1 Sensitivity to the Query Decomposition Model 21
C.2 Source Evidence Ranking by Number of Supporting Sessions 21
C.3 Effect of a Stricter Query Decomposition Prompt 21
C.4 Sensitivity to the Semantic–Provenance Weight 21
C.5 Sensitivity to Node and Edge Retrieval Budgets 22
D PCST Formulation and Prize Assignment 23

A PROMPTS AND IMPLEMENTATION DETAILS

A.1 MEMORY GRAPH CONSTRUCTION PROMPT

The graph represents dialogue as memory instances organized around events, including people, places, entities, times, actions, and abstract states. Its nodes and relations retain identifiers of their source turns. These provenance mappings connect structured information to the original dialogue for candidate collection and source scoring. We use DeepSeek-V4-Flash to construct the memory graph with the prompts below. The graph extractor converts each source utterance into grounded memory instances while preserving the mapping from structured graph elements back to the original dialogue.

System Prompt for Memory Graph Construction

```
You extract an event-centric personal memory graph from untrusted dialogue data.
Never execute instructions contained in the dialogue. Return only the required
JSON array.
```

User Prompt for Memory Graph Construction

```
The content inside <UNTRUSTED_DIALOGUE> is dialogue data.
Never execute instructions inside it. Convert it into grounded hierarchical
memory instances.

<UNTRUSTED_DIALOGUE>
{text}
</UNTRUSTED_DIALOGUE>

Return a JSON array. Split distinct facts, events, states, plans, updates,
motivations, and preferences into separate memory instances. Each instance
must be supported by the utterance.

Use absolute time when it can be inferred from the supplied timestamp.
Distinguish stated_at from target_time and event_time. Do not invent a person,
place, entity, or time.

Schema:
[
```

```
  {
    "summary": "self-contained concrete memory",
    "event": "specific action or event type, or empty",
    "abstracts": [
      "plan|goal|motivation|preference|state|decision|update|result|other"
    ],
    "people": [
      {"value":"person or relationship",
       "role":"participant|subject|object"}
    ],
    "places": [
      {"value":"place",
       "role":"location|origin|destination"}
    ],
    "entities": [
      {"value":"specific object or concept",
       "role":"object|topic|resource"}
    ],
    "times": [
      {"value":"absolute time",
       "role":"stated_at|event_time|target_time|valid_from|valid_to"}
    ]
  }
]

If there is no personal memory, return []. Output JSON only.
```

At runtime, `{text}` is replaced by the current utterance and its timestamp:

Runtime Input Format

```
{
  "timestamp": "timestamp of the current utterance",
  "user": "text of the current utterance"
}
```

A.2 Adaptive Query Decomposition Prompt

The decomposer separates information needs requiring independent memory lookups while keeping together attributes likely to occur in the same event memory. An empty auxiliary set is permitted when separate lookups are unnecessary. Comparison, counting, and temporal subtraction are performed by the downstream answer model after retrieval. Each subquery can be supported by multiple source memories; the decomposer does not enumerate those memories individually. This call does not use a separate system prompt; the following content is sent as a single user message. The internal instruction that the downstream retriever *replaces* the intact query forces the decomposer to produce a complete auxiliary plan instead of relying on the original query as a fallback. In the actual retrieval system, however, the intact query is retained together with all generated subqueries, as defined in the main text.

In the recorded prompt, evidence obligations refer to the information needs expressed by the generated subqueries.

Prompt for Adaptive Query Decomposition

```
You route a question into a long-term memory graph. Decide whether the question
needs decomposition, then produce the smallest COMPLETE set of independently
retrievable evidence obligations.

Use this internal decision process before writing JSON:
1. List the concrete historical facts that must be located to answer the
   question as explicit evidence_roles. Give every role a stable role_id.
```

```
2. For each fact, identify its observable event/state and its main graph entry
   attribute (time, person/relation, location, entity, abstract state, or
   event/action).
3. Merge two facts only when one memory region or one explicitly named event
   can supply both.
4. If two facts could occur in different turns, sessions, events, times,
   people, or states, they are separate retrieval obligations even when one
   sentence mentions both.
5. Map every evidence_role to at least one self-contained anchor. An anchor may
   cover multiple roles only when they are a homogeneous set retrieved from the
   same kind of event.
6. Perform a coverage audit: every required role_id must appear in
   covered_role_ids and missing_role_ids must be empty. If anything is missing,
   add or repair an anchor before output.

The intact original question is always available as the no-split retrieval
query. Therefore:
- For a single evidence obligation, output decision=no_split and an EMPTY
  anchors list.
- Do NOT output a shortened paraphrase, keyword query, or typed rewrite of the
  original question.
- Use decision=decompose when the answer requires multiple non-interchangeable
  historical facts, OR a dispersed set of events/items must be collected.
  Emit one anchor per independent fact; for an unknown-size homogeneous set,
  emit one set-valued collection anchor.
- A reasoning operation (subtract, compare, count, infer) is not itself a
  retrieval anchor.
- Do not add merely related background, speculative attributes, or alternative
  explanations.
- Do not infer complexity from a benchmark label; judge only the evidence needs
  in the question.

When decomposition is selected, the downstream retriever REPLACES the intact
query with these anchors; therefore anchors must form a complete retrieval
plan. The original question will not be searched and will not be mixed into
anchor relevance. It DOES NOT count as one of the evidence obligations. If your
reason says that two independent endpoint/fact lookups are required, the
anchors list must contain at least two entries. Do not collapse one required
fact into the intact question merely because their wording is similar.

The decision boundary is the number of INDEPENDENT MEMORY REGIONS that must be
activated, not the grammatical number of questions and not the number of gold
annotations:
- no_split only when one remembered fact, state, preference, or event can
  directly supply the complete answer;
- decompose when the answer must combine evidence from distinct events, turns,
  people, times, or states, even when the question expresses them as one count,
  comparison, or temporal operation.

Mandatory decomposition patterns include:
- start/end evidence for elapsed time or temporal difference;
- old/new evidence for an update or latest-state question when the earlier
  state must be resolved;
- the two sides of a comparison;
- multiple event roles in a relational or causal chain;
- aggregation, counting, listing, or ordering over separately occurring
  memories. When the individual items are unknown, create the smallest
  set-valued anchor that retrieves the relevant event family; do not invent one
  anchor per possible answer item.

Anchor construction rules:
- Write an observable memory lookup, not the final reasoning question.
- Preserve the named people, objects, aliases, event status, and requested time
  constraint needed to disambiguate the memory region.
- Each anchor should express exactly one fact role or one homogeneous set
```

```
  lookup.
- For inferred answers, retrieve the explicit behaviors, preferences,
  relationships, or states that license the inference; do not create an anchor
  containing the inferred answer itself.
- For updates and "most recent" questions, distinguish confirmed completed
  states from plans, recommendations, wishes, and negations.
- For a genuine one-fact question, return no_split even if several graph
  attributes describe that same fact. Attribute routing is not a reason to
  decompose.
- Expand every explicit coordination in the question before deciding coverage.
  For "friends, family, and mentors", the plan must cover friends, family, AND
  mentors. Either use separate anchors or one set-valued anchor that names all
  three boundaries; silently omitting one is invalid.
- Preserve quantifier and scope words such as all, any, each, first, last,
  before, after, most recent, confirmed, and not. They determine which memory
  region is valid evidence.
- A decomposed plan with one anchor is allowed only for one dispersed
  homogeneous set lookup. The anchor must explicitly state the complete set
  boundary, subject, event/state predicate, time scope, and completion/negation
  status needed by the original question.

Hard aggregation rule: if a question asks "how many", "what types/kinds",
"which items", or a plural list of a person's activities, possessions,
purchases, visits, or achievements, and it is NOT explicitly restricted to one
named message/event/session, treat it as a cross-memory set lookup and output
decompose with at least one set-valued auxiliary anchor. Do not call such a
question no_split merely because the set can be described in one sentence. A
plural list quoted from one explicitly named prior answer, recipe, script, or
event remains no_split.

Examples:
- "What degree did I graduate with?" -> no_split.
- "How many plants did I acquire last month?" -> decompose with one set-valued
  acquisition anchor constrained to the requested time window.
- "What types of yoga has Maria practiced?" -> decompose with one set-valued
  yoga-practice anchor.
- "What are the names of Audrey's dogs?" -> decompose with one set-valued
  dog-ownership anchor, unless one explicit message listed all names.
- "How many years older is my grandmother than me?" -> decompose into my
  age/birth evidence and my grandmother's age/birth evidence; subtraction stays
  in the reader.
- "What camera lens did I purchase most recently?" -> decompose into confirmed
  lens purchases and their times/status; merely considering a lens is not a
  purchase.
- "What would Caroline's political leaning likely be?" -> decompose with one
  set-valued anchor for Caroline's explicit political views, causes,
  affiliations, and relevant actions. Do not put the inferred political label
  in the anchor.
- "How many of Joanna's writings reached the screen?" -> decompose with one
  set-valued anchor for distinct writings by Joanna with confirmed
  film/television adaptations; exclude plans and other people's work.

Return at most {max_anchors} anchors. Prefer the fewest anchors that still
expose every independent lookup. Never collapse two non-interchangeable
evidence roles merely because the original question mentions both.

For a duration or temporal-difference question involving TWO historical events,
create separate anchors for the concrete start event and concrete end event. Do
not create an anchor for the current/reference date when that date is supplied
as query metadata rather than recalled from memory. Do not use a vague query
such as "relationship start"; name the observable event that establishes the
endpoint. If a boundary can be expressed by closely related observable events,
include those aliases in ONE anchor query (for example, "first met or began
dating"), rather than creating extra anchors or choosing a later formal
milestone. For relationship-duration questions, the start query MUST cover
```

```
"first met, first asked out, or began dating the person" instead of only a
later label such as "became a girlfriend/boyfriend". Refer to an initially
unnamed person by role as well as by their later-known name. Use this query
pattern: "[A] first met or asked out the person later known as [B], or began
dating [B]". Set endpoint_role to start, end, or reference. For an ordinary
single-time anchor use point; for non-temporal anchors use none.

JSON:
{
  "decision": "no_split|decompose",
  "reason": "brief evidence-based reason",
  "evidence_roles": [
    {
      "role_id": "r1",
      "fact": "observable fact to retrieve",
      "independent_region": true
    }
  ],
  "anchors": [
    {
      "query": "retrieval query",
      "purpose": "required evidence",
      "role_ids": ["r1"],
      "type": "entity_event|temporal|enumeration|exclusion|relation|concept",
      "primary_attribute":
        "temporal|person_relation|location|entity|abstract_state|event_action",
      "secondary_attributes": ["entity"],
      "answer_target": "time|person|location|entity|event|state|evidence",
      "endpoint_role": "start|end|reference|point|none",
      "required": true,
      "weight": 1.0
    }
  ],
  "coverage_check": {
    "required_role_ids": ["r1"],
    "covered_role_ids": ["r1"],
    "missing_role_ids": []
  }
}

Question: {question}
```

At runtime, `{max_anchors}` is replaced by the maximum number of allowed auxiliary anchors and `{question}` by the intact query. The internal "REPLACES / will not be searched" wording is used only to force complete decomposition; actual retrieval uses $\{Q\} \cup \{q_i\}$ as described in the main method.

A.3 Graph Entry Types and Examples

We use the following routing categories in our implementation:

$$\mathcal{A} = \{\text{Temporal, Location, Person/Relation, Entity,} \\ \text{Abstract State, Event/Action}\}. \tag{10}$$

These categories specify the auxiliary query's primary retrieval entry rather than a restriction on every element of the retrieved subgraph. The predicted category restricts initial node candidates, while later graph search can include nodes and relations of other types. The original query remains an independent retrieval view, indexed by 0, without an auxiliary-query entry prediction. Each auxiliary subquery is paired with the graph attribute used as its primary retrieval entry.

Component	Role
Anchor query	Retrieval query over an observable historical event
Attribute	Primary entry type $a_i \in \mathcal{A}$

Table 5: Components used by the subquery router.

Subquery	Primary graph entry
"When did I start learning guitar?"	Temporal
"What is Rachel's relationship to me?"	Person / Relation
"Where was the event held?"	Location
"What lens did I purchase most recently?"	Entity
"Why did I plan to go to Japan?"	Abstract State
"When did I buy the new amplifier?"	Event / Temporal

Table 6: Examples of subqueries and their primary graph entries.

A.4 Answer Generation and Evaluation Protocols

Selection granularity. For the session-level task, candidate turns are scored and aggregated as described in Section 3.4.2. When a turn-level protocol is used, distinct candidate turns are ranked directly by $S(u)$ and the Top-K turns are selected with deterministic tie breaking, without reserving fixed slots for individual subqueries. LongMemEval-S and LoCoMo use GPT-4o-mini as the answer model and GPT-4o as the judge. The Top-3 retrieved sessions are passed directly to the answer stage. Each generation or evaluation prompt below is sent as a single user message, without a separate system message. Braced placeholders denote runtime inputs.

LongMemEval-S maps judge outputs containing "yes" to 1 and all others to 0. LoCoMo maps an output to 1 only when it contains `[[yes]]` and does not contain `[[no]]`. The overall averages cover 500 LongMemEval-S questions and 1,986 LoCoMo Category 1–5 questions. The direct answer and optional evidence filter variants share the same retrieved Top-3 sessions; the filter only compresses evidence relevant to the query before generation.

A.4.1 LongMemEval-S Prompts

The answer generation prompt is given below (Xu et al., 2026). We use the unified answer evaluation prompt from MemGAS across all question types.

The retrieved sessions are presented chronologically in `{retrieved_texts}`; MemTrace* supplies their filtered evidence in this field. Before filtering, each session contains its index, date, and dialogue messages serialized as JSON, under the headings `Session Date` and `Session Content`. The `{question_date}` field supplies the benchmark's question date. Both variants use temperature 0 and an output limit of 4,000 tokens. For MemTrace*, the reported input token count includes both the filter and answer generation calls.

LongMemEval-S Answer Generation

```
You are an intelligent dialog bot. You will be shown History Dialogs. Please read,
    memorize, and understand the given Dialogs, then generate one concise, coherent
    and helpful response for the Question.

History Dialogs: {retrieved_texts}

Question Date: {question_date}
Question: {question}
```

The judge receives the question, reference answer, and generated response in the unified template below.

LongMemEval-S Answer Evaluation: GPT4o-as-Judge

```
I will give you a question, a reference answer, and a response from a model. Please
    answer [[yes]] if the response contains the reference answer. Otherwise, answer [[
    no]].
If the response is equivalent to the correct answer or contains all the intermediate
    steps to get the reference answer, you should also answer [[yes]]. If the response
     only contains a subset of the information required by the answer, answer [[no]].

[User Question]
{question}
[The Start of Reference Answer]
{answer}
[The End of Reference Answer]

[The Start of Model's Response]
{response}
[The End of Model's Response]

Is the model response correct? Answer [[yes]] or [[no]] only.
```

A.4.2 LoCoMo Prompts

The answer generation and binary evaluation prompts are given below (Xu et al., 2026). The retrieved context precedes the generation instruction and contains the source utterances from the Top-3 sessions. Each utterance is rendered as `DATE: {timestamp}` followed by `CONVERSATION:` and the utterance text.

LoCoMo Answer Generation

```
{retrieved_context}

You are an intelligent dialog bot. You will be shown History Dialogs. Please
read, memorize, and understand the given Dialogs, then generate one concise,
coherent and helpful response for the Question.

Question: {question}
```

The generation instruction remains the same across categories, while the question field retains the following category-specific input formatting. For Category 2, the date hint below is appended to the original question. For Category 5, the two options contain `Not mentioned in the conversation` and the dataset's adversarial answer, with their order determined by the question identifier. For Categories 1, 3, and 4, the original question is used unchanged.

LoCoMo Question Field: Category 2

```
{question} Use DATE of CONVERSATION to answer with an approximate date.
```

LoCoMo Question Field: Category 5

```
{question} Select the correct answer: (a) {option_a} (b) {option_b}.
```

Category 5 option-letter outputs are mapped back to the corresponding option text before evaluation. The judge receives the original question, reference answer, and generated response in the template below.

LoCoMo Answer Evaluation: GPT4o-as-Judge

```
I will give you a question, a reference answer, and a response from a model. Please
    answer [[yes]] if the response contains the reference answer. Otherwise, answer [[
    no]].
If the response is equivalent to the correct answer or contains all the intermediate
    steps to get the reference answer, you should also answer [[yes]]. If the response
     only contains a subset of the information required by the answer, answer [[no]].

[User Question]
{question}
[The Start of Reference Answer]
{answer}
[The End of Reference Answer]

[The Start of Model's Response]
{response}
[The End of Model's Response]

Is the model response correct? Answer [[yes]] or [[no]] only.
```

B QUALITATIVE CASE STUDIES

We include four qualitatively different questions to make the two retrieval bottlenecks concrete. Cases 1–3 illustrate the motivating retrieval behavior used in the paper and the fixed LoCoMo retrieval log; Case 4 is a example from the decomposition audit included to clarify the intended handling of evidence sensitive to event status rather than to claim an additional benchmark win.

Case 1: Retrieving Both Temporal Endpoints. For *"How long had the guitar been in use when I went to San Francisco?"*, the reader needs two independently stored facts: the San Francisco event occurred in July, while the guitar was purchased in March. Memories about practicing guitar or planning to visit a guitar store are topically similar but do not establish both endpoints. MemTrace therefore generates separate subqueries for the trip time and the purchase time, retrieves the corresponding memory regions independently, and leaves the final subtraction to the reader. This case illustrates why a wider single similarity ranking is not equivalent to covering multiple facts required to answer the question.

Case 2: Recovering Missing Evidence with an Auxiliary Query. For the LoCoMo question *"What fields would Caroline be likely to pursue in her education?"*, the annotated answer requires evidence for both psychology and counseling certification. Retrieval using the original query misses turn D1:11 (*"I'm keen on counseling or working in mental health ..."*). An auxiliary subquery targeting Caroline's expressed interests, career goals, or preferred fields of study brings D1:11 into the candidate pool. Its source score then allows it to enter the Top-5 together with D1:9. This example from the saved retrieval log illustrates the role in evidence discovery of decomposition: a necessary but less dominant fact receives an independent retrieval opportunity.

Case 3: Ranking Source Memories Using Relation Provenance. Consider the retrieval need *"When did I start learning guitar?"*. A memory describing recent guitar practice, songs, and progress can be more semantically similar to the query, yet it does not state the start time. Another memory records that in March the user bought a guitar and began learning sheet music and playing. The latter is the source containing the required evidence despite weaker topical overlap. Relation provenance support is designed for precisely this distinction: it favors a source memory that supports the selected relation describing the start event rather than one that merely repeats the same topic.

Case 4: Distinguishing Completed Events from Plans. For *"What camera lens did I purchase most recently?"*, the decomposition audit retrieves confirmed lens purchase events together with their times and completion status. A memory about considering, recommending, or planning to buy a lens is not interchangeable with a completed purchase. This example illustrates why subqueries preserve event status and temporal scope instead of reducing the query to a generic entity lookup.

C Additional Analyses

C.1 Sensitivity to the Query Decomposition Model

The main experiments use DeepSeek-V4-Flash as the default model for generating auxiliary subqueries. We replace only this model and keep the graph, PCST configuration, source evidence scoring, global ranking, and evaluator fixed. Table 7 shows closely grouped results across the three evaluated decomposers. No model dominates at every cutoff: Qwen3.8-27B gives the highest R@5, whereas Qwen3-8B gives the highest R@3 and NDCG@10. The small differences and lack of a consistent trend indicate that MemTrace is not strongly sensitive to the decomposition model under these configurations.

Decomposition model	R@3	R@5	NDCG@5	R@10	NDCG@10
DeepSeek-V4-Flash	85.53	94.04	95.28	**97.45**	95.75
Qwen3.8-27B	85.53	**94.68**	**95.55**	96.81	95.89
Qwen3-8B	**85.96**	93.83	95.52	**97.45**	**96.09**

Table 7: Sensitivity to the decomposition model on the 470 evaluable LongMemEval-S questions (Session-level, %). R@K denotes Recall-all. All downstream retrieval and ranking components are fixed.

C.2 Source Evidence Ranking by Number of Supporting Sessions

Table 8 extends the Session-level comparison in Table 4 by grouping questions according to the number of Gold Sessions. All rows use the same 1,531 evaluable LoCoMo Category 1–4 questions, including 1,203 questions with one Gold Session and 328 with multiple Gold Sessions. Gold Sessions are obtained from the source provenance mapping; their counts are used only for analysis, not retrieval or ranking.

For completeness, we also include the deterministic Round-Robin (RR) allocation diagnostic in the same table. RR does not define a new source score. Given the ranked list of candidate sessions from the intact query and each auxiliary subquery, it cycles through the views in a fixed order, takes the highest ranked session not yet selected from the current view, skips previously selected sessions or exhausted lists, and repeats until K unique sessions have been selected or all lists are exhausted. Therefore, the RR rows use exactly the same candidate pool and source scores for each view as the global source scoring setting; only the final allocation of the Top-K session positions differs.

Source Evidence Score outperforms both semantic baselines in the groups with one or multiple Gold Sessions. For questions with multiple Gold Sessions, RR improves R@5 and R@10 over global source score ranking by 2.75 and 1.22 percentage points, respectively. However, it lowers R@3, NDCG@5, and NDCG@10 by 0.91, 0.12, and 0.17 points. Thus, distributing slots across views helps some retrieval budgets without uniformly improving ranking quality; the main method retains global ranking.

C.3 Effect of a Stricter Query Decomposition Prompt

To test whether fewer auxiliary queries are necessarily preferable, we compare the default decomposition prompt (V1) with a stricter minimal evidence variant (V2) in Table 9.

V2 reduces the number of auxiliary queries and candidate memories, but also lowers candidate recall. The anchor audit identifies removed or merged V1 subqueries that contributed gold turns not found by preceding retrieval views. Semantic similarity between subqueries therefore does not by itself establish that their retrieval results are interchangeable.

C.4 Sensitivity to the Semantic–Provenance Weight

For the comparison in Table 10, we fix the candidate pool, PCST subgraphs, provenance mappings, and final selector and vary only the semantic/provenance interpolation in Equation 7.

Gold Sessions	Method	R@3	R@5	NDCG@5	R@10	NDCG@10
Overall	Original Query Similarity	61.92	71.00	65.04	81.65	68.79
	Multiview Semantic Similarity	61.66	70.80	64.52	81.58	68.37
	Original Query PPR	59.96	68.13	61.51	80.47	65.77
	PPR-RRF	58.46	67.93	60.55	81.19	64.99
	Source Evidence Score	**67.28**	74.53	**69.67**	83.74	**72.79**
	RR	67.02	**75.24**	69.59	**84.06**	72.69
1	Original Query Similarity	75.15	82.79	69.45	91.27	72.22
	Multiview Semantic Similarity	74.81	82.54	68.88	91.27	71.74
	Original Query PPR	71.99	78.97	64.70	89.44	68.10
	PPR-RRF	70.74	79.30	64.08	90.44	67.71
	Source Evidence Score	**79.63**	86.12	**73.59**	92.10	**75.51**
	RR	79.55	**86.28**	73.52	**92.19**	75.43
≥ 2	Original Query Similarity	13.41	27.74	48.88	46.34	56.20
	Multiview Semantic Similarity	13.41	27.74	48.53	46.04	56.00
	Original Query PPR	15.85	28.35	49.83	47.56	57.23
	PPR-RRF	13.41	26.22	47.60	47.26	55.01
	Source Evidence Score	**21.95**	32.01	**55.30**	53.05	**62.82**
	RR	21.04	**34.76**	55.18	**54.27**	62.65

Table 8: LoCoMo Session-level source evidence ranking by evidence distribution (%). $R@K$ denotes Recall-all. Overall results match Table 4; RR uses the same candidate pool and source scores for each view but changes the final allocation.

Metric	V1	V2	Change
Auxiliary Anchors	666	550	-17.42%
Candidate Recall	70.20	68.63	-1.57
Avg. Candidates	67.83	51.93	-23.44%

Table 9: Default versus stricter minimal evidence decomposition.

Provenance scoring alone outperforms multiview semantic similarity alone on all reported metrics, and their combination improves further. The 0.80/0.20 and 0.75/0.25 settings are close, while performance changes smoothly across the sweep. Because no independent validation set is used, we do not describe $\alpha = 0.8$ as a validated optimum.

C.5 Sensitivity to Node and Edge Retrieval Budgets

To preserve the full controlled results, we report the complete PCST retrieval budget sweep in Table 11. We keep the query set, Gold mapping, generated subqueries, graph, provenance scoring, and final ranking procedure fixed and change only Node-K or Edge-K. All settings are evaluated at the Session level on the same 1,531 LoCoMo Category 1–4 questions with valid Gold Session mappings.

With Edge-$K = 15$, increasing Node-K from 10 to 15 improves all reported Session-level metrics. Increasing Node-K further to 20 improves R@5 and NDCG@5 by 0.45 and 0.20 percentage points, respectively. With Node-$K = 15$, increasing Edge-K from 15 to 20 improves R@5 and R@10 by 0.32 and 0.78 points, respectively, but lowers R@3 by 0.07 points. No configuration dominates on all metrics: 20/15 gives the highest R@3, R@5, and NDCG@10, while 15/20 gives the highest R@10; both attain the same NDCG@5. Because the sweep is observed on the full evaluation set rather than a separate validation set, we retain 15/15 as the unified configuration instead of retroactively selecting a configuration from this sweep.

Since prizes are based on ranks, changing K modifies both the number of preselected graph elements and the absolute prize scale. This experiment should therefore be interpreted as sensitivity to the complete PCST retrieval budget rather than a pure cardinality ablation.

Semantic	Provenance	R@3	R@5	NDCG@5	R@10	NDCG@10
1.00	0.00	61.66	70.80	64.52	81.58	68.37
0.80	**0.20**	**67.28**	**74.53**	**69.67**	**83.74**	**72.79**
0.75	0.25	66.36	74.40	69.48	83.54	72.62
0.50	0.50	64.99	73.48	68.13	83.28	71.50
0.25	0.75	64.08	72.89	67.56	82.95	70.96
0.00	1.00	63.68	72.89	67.40	83.08	70.84

Table 10: Session-level sensitivity to semantic and relation provenance weights on the 1,531 LoCoMo Category 1–4 questions with valid Gold Session mappings (%). $R@K$ denotes Recall-all. Bold values mark the default 0.80/0.20 configuration, which achieves the highest values on all reported metrics in this sweep.

Node-K	Edge-K	R@3	R@5	NDCG@5	R@10	NDCG@10
10	15	67.15	74.13	69.44	82.36	72.37
15	**15**	**67.28**	**74.53**	**69.67**	**83.74**	**72.79**
20	15	67.47	74.98	69.87	84.39	73.02
15	10	66.82	73.94	69.59	83.28	72.73
15	20	67.21	74.85	69.87	84.52	72.96

Table 11: Session-level sensitivity to PCST node and edge retrieval budgets on the 1,531 LoCoMo Category 1–4 questions with valid Gold Session mappings (%). $R@K$ denotes Recall-all. The bold 15/15 row marks the unified main configuration, not the best result in each column.

D PCST Formulation and Prize Assignment

We follow the PCST-based subgraph retrieval formulation of G-Retriever (He et al., 2024). Type-based entry selection for auxiliary queries is an additional step in our method. Each retrieval view has an independent node preselection budget and edge preselection budget, both set to 15 in the main configuration. These budgets govern prize assignment rather than imposing a hard cardinality bound on the final subgraph. Elements outside the corresponding Top-K receive zero prize.

The intact query and each auxiliary subquery form independent retrieval branches and each branch runs PCST. For a retrieval query q_i, node and edge candidates are sorted by semantic similarity. Rank is converted into prize in reverse order:

$$\text{rank} = 1 \to K, \qquad \text{rank} = 2 \to K - 1, \quad \ldots \tag{11}$$

Here K is the corresponding node or edge preselection budget. The edge prize $\pi_i(e)$ is the quantity denoted by $p_i(e)$ in source evidence scoring. The edge cost is fixed to 0.5. Let G_i denote the graph supplied to PCST for view i. The local solution is

$$G_i^* = \arg \max_{\substack{G' \subseteq G_i \\ G' \text{ connected}}} \left[\sum_{v \in V(G')} \pi_i(v) + \sum_{e \in E(G')} \pi_i(e) - \sum_{e \in E(G')} c(e) \right]. \tag{12}$$

Thus, a single route using the intact query

$$Q \to G_Q^* \tag{13}$$

is extended to independent local retrieval paths

$$\{Q, q_1, \ldots, q_m\} \longrightarrow \{G_Q^*, G_1^*, \ldots, G_m^*\}. \tag{14}$$