

HiGMem: A Hierarchical and LLM-Guided Memory System for Long-Term Conversational Agents

Shuqi Cao^{1†}, Jingyi He^{2†}, Fei Tan^{1*}

¹East China Normal University, Shanghai, China

²Shanghai Jiao Tong University, Shanghai, China

ftan@mail.ecnu.edu.cn

Abstract

Long-term conversational large language model (LLM) agents require memory systems that can recover relevant evidence from historical interactions without overwhelming the answer stage with irrelevant context. However, existing memory systems, including hierarchical ones, still often rely solely on vector similarity for retrieval. It tends to produce bloated evidence sets: adding many superficially similar dialogue turns yields little additional recall, but lowers retrieval precision, increases answer-stage context cost, and makes retrieved memories harder to inspect and manage. To address this, we propose HiGMem (Hierarchical and LLM-Guided Memory System), a two-level event-turn memory system that allows LLMs to use event summaries as semantic anchors to predict which related turns are worth reading. This allows the model to inspect high-level event summaries first and then focus on a smaller set of potentially useful turns, providing a concise and reliable evidence set through reasoning, while avoiding the retrieval overhead that would be excessively high compared to vector retrieval. On the LoCoMo10 benchmark, HiGMem achieves the best F1 on four of five question categories and improves adversarial F1 from 0.54 to 0.78 over A-Mem, while retrieving an order of magnitude fewer turns. Code is publicly available at <https://github.com/ZeroLoss-Lab/HiGMem>.

1 Introduction

As LLMs evolve into autonomous agents, memory systems have become essential for long-term autonomy (Zhang et al., 2024; Wang et al., 2024), transforming historical data into actionable insights to maintain persona consistency and leverage past interactions for contextually-aware responses. Effective memory allows an agent to ground its re-

sponses in historical context, transforming raw interaction data into a coherent understanding of its world and relationships.

However, existing memory systems, including hierarchical ones, still rely predominantly on vector similarity for retrieval (Packer et al., 2023; Xu et al., 2025). While effective at casting a wide net, this approach tends to produce bloated evidence sets: once the most relevant memories have been recalled, adding more superficially similar fragments yields diminishing gains in recall but steadily erodes retrieval precision, inflates the context fed to the answer stage, and makes the evidence set harder to inspect and manage (Park et al., 2023; Lewis et al., 2020). The root cause is that vector similarity alone cannot judge whether a memory is truly worth reading as it lacks a mechanism for reasoning over different levels of abstraction to actively assess which fine-grained details actually contribute to answering a query. Consequently, developing retrieval strategies that can deliver concise, high-precision evidence sets remains a pivotal research objective.

To address these challenges, we introduce HiGMem, a hierarchical and LLM-guided memory system for long-term conversational agents. HiGMem tackles the core limitation of vector-only retrieval through two complementary designs. First, a two-level event-turn architecture organizes memories by abstraction level, from high-level thematic summaries to granular conversational details, providing the structural foundation for multi-granularity reasoning. Second, an LLM-guided Hierarchical Retrieval strategy leverages event summaries as semantic anchors, enabling the LLM to actively predict which fine-grained turns are worth reading rather than passively returning all similar fragments. Empirically, on the LoCoMo10 benchmark (Maharana et al., 2024), HiGMem achieves the best F1 on four of five question categories while retrieving an order of magnitude fewer turns than

[†]Equal contribution.

^{*}Corresponding author.

the strongest baseline, demonstrating that hierarchical LLM-guided retrieval can provide a concise and reliable evidence set through reasoning, while avoiding the retrieval overhead that would be excessively high compared to vector retrieval.

Our contributions are twofold: (i) We propose a retrieval paradigm for long-term conversational memory that combines hierarchical abstraction with LLM-guided filtering, enabling the system to deliver concise and reliable evidence sets through reasoning while avoiding the prohibitive cost of applying LLM reasoning to every stored memory. (ii) We implement this paradigm as HiGMem, an open-source memory system evaluated on the LoCoMo10 benchmark. HiGMem achieves the best F1 on four of five question categories while retrieving an order of magnitude fewer turns, providing a reproducible baseline for future research on long-term conversational memory.

2 Related Work

Long-term conversational agents depend on both strong multi-turn dialogue modeling and effective memory retrieval over long interaction histories. Recent training-time methods such as ReSURE (Du et al., 2025) and daDPO (Zhang et al., 2025) improve multi-turn conversational ability through more reliable supervision and distribution-aware distillation. Our work focuses on the memory side of this broader problem: retrieval-time evidence localization for long-term conversational agents.

Memory-augmented LLM agents such as MemoryBank (Zhong et al., 2024), MemGPT (Packer et al., 2023), and A-Mem (Xu et al., 2025) extend long-horizon interaction by storing past experiences as external memories and retrieving them at inference time. However, systems that rely solely on vector-similarity retrieval tend to return broad evidence sets once the memory store grows over extended conversations. Once the most relevant evidence has been recalled, adding more semantically similar memories often brings diminishing retrieval precision, inflating context cost, and making evidence harder to inspect and manage.

Recent work has explored more structured memory organization. RAPTOR (Sarathi et al., 2024) uses recursive summaries for multi-granularity retrieval, and concurrent work such as H-MEM (Sun and Zeng, 2025) introduces hierarchical memory routing. These studies highlight the value of structure, but prior evidence remains limited on

whether long-term conversational memory systems can maintain high retrieval recall while passing only a compact set of fine-grained evidence to the answer model.

3 The HiGMem System

HiGMem is designed to address a central trade-off in long-term conversational memory retrieval: the system should recover sufficient evidence for a query, but should do so with a compact and precise evidence set rather than a large pool of semantically similar memories. We therefore pursue three objectives simultaneously: high evidence recall, high retrieval precision, and manageable token cost. To achieve this, HiGMem combines a two-level event-turn memory architecture with LLM-guided retrieval. The event layer provides concise semantic anchors that allow the LLM to inspect memory at low cost, while the turn layer preserves the fine-grained dialogue evidence required for downstream generation. Based on these event summaries, the LLM predicts which related turns are worth reading, producing a smaller and more reliable evidence set than retrieval based solely on vector similarity.

For clarity, in the remainder of this section we use *Turn* node to denote HiGMem’s structured memory unit built from a dialogue turn and its metadata, and *Event* node to denote the higher-level unit that groups related *Turn* nodes.

3.1 Hierarchical Memory Architecture

HiGMem features a streamlined two-level hierarchical architecture that organizes information across different granularities of abstraction. A critical component of this design is the establishment of robust bidirectional links between layers, ensuring both bidirectional interoperability and data provenance traceability. At the foundation lies the turn layer, which captures raw interaction details at the finest granularity. Here, each *Turn* node represents a raw dialogue turn enriched with LLM-generated metadata (e.g., keywords, tags, timestamp and context), serving as the primary evidence base. Conversely, the event layer operates at the top level, grouping relevant *Turn* nodes into coherent narrative units. This layer includes a short summary and a structured fact sheet. The fact sheet orchestrates the timeline and specific *Event* nodes by maintaining explicit index links to the corresponding *Turn* nodes in the bottom layer.

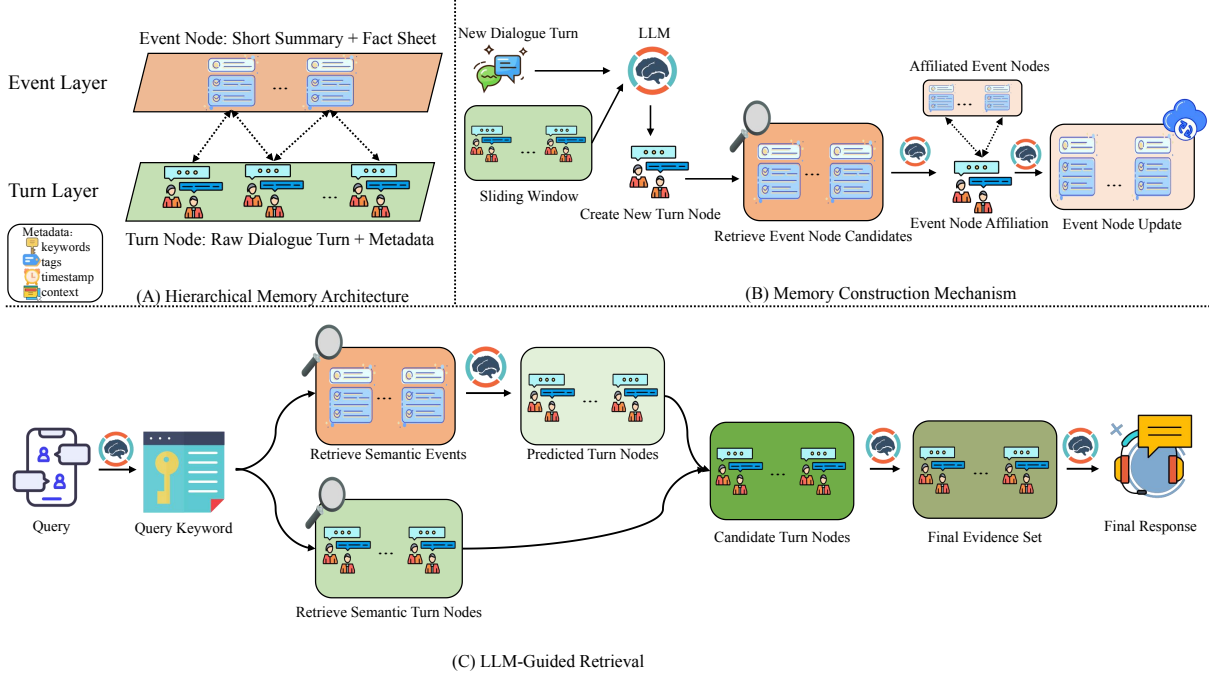


Figure 1: The overall architecture of HiGMem.

3.2 Memory Construction Mechanism

HiGMem supports real-time, automated memory updates. When new dialogue turn D_t is ingested, the system executes an automatic update mechanism.

Turn Analysis. First, an LLM analyzes the incoming dialogue turn D_t within the context of local sliding window $\mathcal{W}_t = \{T_{t-m}, \dots, T_{t-1}\}$ (m denotes the sliding window size) to extract metadata. Metadata \mathcal{M}_t includes keywords, tags, timestamp and context. After analyzing, new *Turn* node T_t consists of dialogue turn D_t and metadata \mathcal{M}_t . This transformation can be formalized as:

$$D_t \xrightarrow{\text{LLM}(\cdot|\mathcal{W}_t)} \mathcal{M}_t, \quad (1)$$

$$T_t = (D_t, \mathcal{M}_t), \quad (2)$$

where the arrow in Eq. (1) denotes the generative transformation by the LLM.

Event Affiliation and Update. Simultaneously, the system determines the affiliation of T_t at the event layer. We employ a text encoder to compute the dense vector representation of T_t , denoted as e_t . Subsequently, we calculate similarity score $s_{t,j}$ between T_t and each existing *Event* node E_j in the global set of *Event* nodes \mathcal{E} using cosine similarity:

$$s_{t,j} = \frac{e_t \cdot e_{E_j}}{\|e_t\| \|e_{E_j}\|}, \quad \forall E_j \in \mathcal{E}, \quad (3)$$

where e_{E_j} represents embedding vector of *Event* node E_j . We retrieve the top- k_{event} most relevant events to construct candidate set $\mathcal{E}_{\text{cand}}$. The set of *Event* nodes to which T_t is affiliated \mathbf{E}^* is identified by an LLM-based decision function:

$$\mathbf{E}^* = f_{\text{affiliate}}(T_t | \mathcal{E}_{\text{cand}}), \quad (4)$$

where $\mathbf{E}^* \subseteq \mathcal{E}_{\text{cand}}$ represents a subset of *Event* nodes. Subsequently, for each *Event* node $E \in \mathbf{E}^*$, we perform an adaptive update based on the scale of the interaction. Specifically, based on *Event* node volume $|E|$ (i.e., the number of linked *Turn* nodes in *Event* node E) and threshold τ , we fully refresh both the summary and fact sheet if $|E| < \tau$, but only append the new entry if $|E| \geq \tau$. Finally, to ensure traceability, we update link set \mathcal{L}_E for each affiliated *Event* node by appending new link l between E and T_t :

$$\mathcal{L}_E = \mathcal{L}_E \cup \{l\}, \quad \forall E \in \mathbf{E}^*. \quad (5)$$

In all experiments, sliding window size for turn analysis is fixed to $m = 5$, and adaptive *Event* node update threshold is fixed to $\tau = 10$.

3.3 LLM-Guided Retrieval

To answer query Q , HiGMem employs an LLM-guided retrieval strategy. The system first analyzes Q to generate query keyword q_{kw} : $q_{\text{kw}} = \text{LLM}(Q)$. Based on Eq. (3), query q_{kw} simultaneously retrieves the k_{turn} most relevant *Turn* nodes

($\mathcal{T}_{\text{semantic}}$) and the k_{event} most relevant *Event* nodes ($\mathcal{E}_{\text{semantic}}$) from the turn and event layers, respectively. Uniquely, HiGMem utilizes $\mathcal{E}_{\text{semantic}}$ as semantic anchors for further discovery. The LLM evaluates constituent *Turn* nodes \mathcal{T}_E which are linked to each *Event* node E in $\mathcal{E}_{\text{semantic}}$, performing reasoning to select specific *Turn* nodes capable of answering the query. This process yields inference-based set $\mathcal{T}_{\text{pred}}$:

$$\mathcal{T}_{\text{pred}} = \bigcup_{E \in \mathcal{E}_{\text{semantic}}} \text{Predict}(\mathcal{T}_E \mid Q, E). \quad (6)$$

The system integrates $\mathcal{T}_{\text{semantic}}$ and $\mathcal{T}_{\text{pred}}$, and then employs an LLM-based filtering mechanism to construct final evidence set, denoted by $\mathcal{T}_{\text{final}}$, which consists of *Turn* nodes:

$$\mathcal{T}_{\text{cand}} = \mathcal{T}_{\text{semantic}} \cup \mathcal{T}_{\text{pred}}, \quad (7)$$

$$\mathcal{T}_{\text{final}} = \text{Filter}(\mathcal{T}_{\text{cand}} \mid Q), \quad (8)$$

where $\text{Filter}(\cdot \mid Q)$ denotes the filtering process. Finally, *Turn* nodes in the final evidence set $\mathcal{T}_{\text{final}}$ and query Q are fed into the LLM to generate a response.

4 Experiments

4.1 Experimental Setup

Dataset. We evaluate HiGMem on LoCoMo10, a subset of 10 longest conversations with high-quality annotations selected by the authors of LoCoMo (Maharana et al., 2024) from the full LoCoMo dataset, with an average of 587 turns per conversation. LoCoMo10 is particularly suitable for our setting because it contains significantly longer multi-session dialogues than existing conversational datasets (Xu et al., 2022; Jang et al., 2023), requiring models to recover evidence over extended interaction histories. Based on its question answering (QA) benchmark, we evaluate memory retrieval across five categories: Single-Hop, Multi-Hop, Temporal, Open-Domain, and Adversarial questions.

Baselines. We compare HiGMem against ReadAgent (Lee et al., 2024), MemoryBank (Zhong et al., 2024), A-Mem (Xu et al., 2025), and a full-context GPT-4o-mini baseline following prior LoCoMo10 evaluations. We do not include H-MEM (Sun and Zeng, 2025) in the main quantitative comparison because we could not identify an official public

implementation, and key answer-stage details are insufficient for fair reproduction.

Metrics. We report F1 score (Rajpurkar et al., 2016) and retrieval metrics, including Avg K, Precision@K, and Recall@K. Here, K denotes the number of turns in the final evidence set provided to the answer-generation LLM for each question. Precision@K is the fraction of retrieved turns that match gold evidence turns, while Recall@K is the fraction of gold evidence turns recovered by the final evidence set. Since a question may require one or more gold evidence turns, we compute Precision@K and Recall@K at the question level and report their macro-average over questions, rather than pooling evidence turns across all questions.

Implementation. For a controlled comparison, all LLM-based systems use GPT-4o-mini (OpenAI, 2023), and all methods use the same final QA prompt for answer generation. HiGMem and A-Mem use the same all-MiniLM-L6-v2 (Reimers and Gurevych, 2019) encoder for vector retrieval. For HiGMem, we set $k_{\text{turn}} = 10$ and $k_{\text{event}} = 10$ for turn- and event-level retrieval. We summarize the main prompt families in Appendix C; complete prompt templates, implementation constants, and evaluation scripts are provided in the released code.

4.2 Results and Analysis

Table 1 summarizes performance on the LoCoMo10 benchmark (Maharana et al., 2024). HiGMem achieves the best F1 on four of five categories, including Multi-Hop, Open-Domain, Single-Hop, and Adversarial questions. Its Temporal F1 is slightly lower than A-Mem, suggesting that the current event-level abstraction may still weaken some fine-grained chronological cues. Overall, the results show that HiGMem is effective in supporting question answering over long conversational histories.

To analyze the source of the F1 score gains, Table 2 directly evaluates the quality of the retrieved evidence set. Compared with A-Mem, HiGMem retrieves an order of magnitude fewer turns on average (8.09 vs. 99.84), while achieving much higher Precision@K (0.1909 vs. 0.0101) and comparable Recall@K (0.7241 vs. 0.7502). This supports our central claim: HiGMem does not simply increase context size to improve recall, but instead produces a compact and more precise evidence set.

Table 3 further clarifies the token cost trade-off. HiGMem uses more GPT-4o-mini tokens during memory construction and retrieval, but substan-

Method	Multi-Hop	Temporal	Open-Domain	Single-Hop	Adversarial	Ranking
Base LLM	0.25	0.39	0.12	0.44	0.30	2.2
ReadAgent	0.09	0.13	0.05	0.10	0.10	4.2
MemoryBank	0.05	0.10	0.06	0.07	0.07	4.8
A-Mem	0.27	0.39	0.10	0.42	0.54	2.2
HiGMem (Ours)	0.31	0.34	0.15	0.49	0.78	1.2

Table 1: Experimental results on the LoCoMo10 dataset across five question categories, reported in F1 score. Ranking is the average category-wise rank across the five question categories, where lower is better. The best performance is highlighted in bold.

Method	Avg K	Precision@K	Recall@K
A-Mem	99.84	0.0101	0.7502
HiGMem	8.09	0.1909	0.7241

Table 2: Retrieval effectiveness on LoCoMo10. Avg K denotes the average number of turns in the final evidence set provided to the answering-generation LLM. HiGMem achieves comparable recall with a much smaller and more precise evidence set.

tially reduces the expensive GPT-5 answer-stage input from 25.38M to 1.62M tokens in a hybrid deployment setting. As a result, the estimated total cost decreases from \$17.43 to \$6.43, about a $2.7\times$ reduction. Table 4 reports the average time overhead. HiGMem requires roughly $2.4\times$ more time for the memory construction stage and $1.6\times$ more for the question answering stage due to additional LLM reasoning calls, a trade-off consistent with the token-cost shift discussed above.

Additional controlled experiments are reported in Appendix B, including a flat retrieval baseline with LLM selection, fixed-budget retrieval comparisons with A-Mem, a supplementary DialSim generalization study, and a Qwen2.5-3B backbone comparison.

4.3 Ablation Study

To validate the role of the event-turn hierarchy, we remove the event layer and degrade HiGMem to a single-layer memory system. As shown in

Method	A-Mem	HiGMem
GPT-4o-mini Tokens	11.81M	54.35M
GPT-5 Answer Tokens	25.38M	1.62M
GPT-4o-mini Cost	\$1.44	\$5.30
GPT-5 Cost	\$15.99	\$1.12
Total Cost	\$17.43	\$6.43

Table 3: End-to-end cost analysis under a hybrid deployment setting, where memory construction and retrieval use GPT-4o-mini, while final answer generation uses GPT-5. HiGMem consumes more tokens before answer generation, but substantially reduces the expensive GPT-5 answer-stage input.

Time Metric	A-Mem	HiGMem (Ours)
Memory Construction (s)	6.38	15.59
Question Answering (s)	5.91	9.42

Table 4: Comparison of computational efficiency in terms of average per-instance time overhead. Memory construction is measured per ingested dialogue turn, while question answering is measured per question.

Method	F1	Recall@K
w/o Hierarchy	0.39	0.55
HiGMem	0.49	0.72

Table 5: Ablation study to verify the effectiveness of the hierarchical structure.

Table 5, removing the hierarchy reduces both F1 and Recall@K. This indicates that the event layer is not merely a storage abstraction: event summaries provide compact semantic anchors that reduce the turn-level search space and help the LLM select more relevant fine-grained evidence.

5 Conclusion

We present HiGMem, a hierarchical memory system for long-term conversational agents that addresses a central retrieval trade-off: recovering sufficient evidence while keeping the evidence set compact and precise. HiGMem combines an event-turn memory architecture with LLM-guided retrieval, using event summaries as semantic anchors to predict which fine-grained turns are worth reading. On LoCoMo10, HiGMem achieves the best F1 on four of five question categories while retrieving far fewer turns than A-Mem, and substantially reduces answer-stage cost in hybrid deployment. These results suggest that LLM-guided evidence selection is an effective direction for building long-term conversational memory systems.

Limitations

HiGMem can reduce downstream answer-stage cost by producing a compact final evidence set,

but it still requires additional LLM calls for memory construction and LLM-guided evidence selection. Therefore, its practical advantage depends on the deployment setting: it is most beneficial when a lightweight model handles memory operations while a more expensive downstream model performs final answer generation. Reducing the time and token overhead of these intermediate LLM calls while preserving evidence precision remains an important direction.

HiGMem also depends on the LLM’s ability to judge relevance from event summaries and candidate turns. Although supplementary experiments with Qwen2.5-3B show a consistent trend on LoCoMo10, the preliminary DialSim study shows lower F1 than A-Mem despite a much smaller evidence set, suggesting that the current event-turn structure is not yet fully optimized for multi-party dialogue. Broader evaluation on real-world, multi-party, noisy, and dynamically evolving conversations is needed to fully assess robustness.

Acknowledgments

This work is supported by the East China Normal University “Artificial Intelligence” Seed Grant Program (40500-20101-222438) and the East China Normal University “Discipline Advancement Program” (40600-515100-25001/002/015).

References

- Yiming Du, Yifan Xiang, Bin Liang, Dahua Lin, Kam-Fai Wong, and Fei Tan. 2025. [ReSURE: Regularizing supervision unreliability for multi-turn dialogue fine-tuning](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 18967–18985, Suzhou, China. Association for Computational Linguistics.
- Jihyoung Jang, Minseong Boo, and Hyoungun Kim. 2023. [Conversation chronicles: Towards diverse temporal and relational dynamics in multi-session conversations](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13584–13606, Singapore. Association for Computational Linguistics.
- Kuang-Huei Lee, Xinyun Chen, Hiroki Furuta, John Canny, and Ian Fischer. 2024. [A human-inspired reading agent with gist memory of very long contexts](#). *arXiv preprint arXiv:2402.09727*.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. [Retrieval-augmented generation for knowledge-intensive nlp tasks](#). *Advances in Neural Information Processing Systems*, 33:9459–9474.
- Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 2024. [Evaluating very long-term conversational memory of llm agents](#). *arXiv preprint arXiv:2402.17753*.
- OpenAI. 2023. [Gpt-4 technical report](#). *arXiv preprint arXiv:2303.08774*.
- Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. 2023. [Memgpt: Towards llms as operating systems](#). *arXiv preprint arXiv:2310.08560*.
- Joon Sung Park, Joseph O’Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. [Generative agents: Interactive simulacra of human behavior](#). In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. [Squad: 100,000+ questions for machine comprehension of text](#). *arXiv preprint arXiv:1606.05250*.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-bert: Sentence embeddings using siamese bert-networks](#). *arXiv preprint arXiv:1908.10084*.
- Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D Manning. 2024. [Raptor: Recursive abstractive processing for tree-organized retrieval](#). In *The Twelfth International Conference on Learning Representations*.
- Haoran Sun and Shaoning Zeng. 2025. [Hierarchical memory for high-efficiency long-term reasoning in llm agents](#). *arXiv preprint arXiv:2507.22925*.
- Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Ji-Rong Wen. 2024. [A survey on large language model based autonomous agents](#). *Frontiers of Computer Science*, 18(6):186345.
- Jing Xu, Arthur Szlam, and Jason Weston. 2022. [Beyond goldfish memory: Long-term open-domain conversation](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5180–5197, Dublin, Ireland. Association for Computational Linguistics.
- Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. 2025. [A-mem: Agentic memory for llm agents](#). *arXiv preprint arXiv:2502.12110*.
- Zeyu Zhang, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Quanyu Dai, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. 2024. [A survey on the memory mechanism of large language model based agents](#). *arXiv preprint arXiv:2404.13501*.

Zhengze Zhang, Shiqi Wang, Yiqun Shen, Simin Guo, Dahua Lin, Xiaoliang Wang, Cam Tu Nguyen, and Fei Tan. 2025. [daDPO: Distribution-aware DPO for distilling conversational abilities](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 15421–15437, Vienna, Austria. Association for Computational Linguistics.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. 2024. [Memorybank: Enhancing large language models with long-term memory](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19724–19731.

A Illustrative Comparison of Retrieval Paradigms

Query: "What kind of car does Evan drive?"
Paradigm 1: Passive Vector Retrieval (e.g., RAG, A-Mem)
<ol style="list-style-type: none"> Reduce: Query is reduced to keywords: {"kind", "car", "Evan", "drive"}. Intent is lost. Search: A single vector search retrieves semantically similar but factually irrelevant memories (e.g., "a two-hour drive"). Result: Failure. The system cannot find the specific "Prius" evidence.
Paradigm 2: LLM-Guided Retrieval (HiGMem)
<ol style="list-style-type: none"> Reason: LLM analyzes the full query, identifying retrieval goals: find a <i>thing</i> ('car') related to an <i>entity</i> ('Evan'). Predict: The LLM evaluates the query against event summaries to identify which linked turns potentially serve as the answer. Locate: Based on these predictions, the system pinpoints the exact evidence within the candidate turns. Result: Success. The system retrieves turns mentioning "my new Prius".

Figure 2: A comparison of retrieval paradigms. Passive Vector Retrieval suffers from semantic drift due to keyword reduction. In contrast, HiGMem’s LLM-guided Retrieval leverages reasoning to predict relevant turns from event summaries and locate exact evidence, ensuring accurate recall.

B Supplementary Experiments

B.1 Flat Retrieval + LLM Selection Baseline

We remove the event layer entirely: flat vector search retrieves the top-100 turns, then applies the same LLM relevance filtering and final answer prompt as HiGMem. This controlled baseline tests whether the gain comes merely from LLM filtering or from using event summaries as semantic anchors.

Task / Metric	Flat+LLM	HiGMem
Multi-Hop	0.29	0.31
Temporal	0.37	0.34
Open-Dom	0.11	0.15
Single-Hop	0.46	0.49
Adversarial	0.73	0.78
Overall F1	0.46	0.49

Table 6: Comparison of task-specific and overall F1 scores between the Flat Retrieval + LLM Selection baseline and HiGMem.

Compared with HiGMem, the Flat+LLM baseline retrieves a substantially larger evidence set (22.71 vs. 8.09 turns on average), but still achieves

Metric	Flat+LLM	HiGMem
Avg K	22.7	8.09
Precision@K	0.065	0.190
Recall@K	0.703	0.724

Table 7: Retrieval efficiency and prompt cost comparison.

lower overall F1 (0.46 vs. 0.49), lower Precision@K (0.065 vs. 0.190), and slightly lower Recall@K (0.703 vs. 0.724). This shows that the event layer is not merely an implementation detail for enabling LLM filtering: Event summaries act as semantic anchors that reduce the turn-level search space and improve evidence precision.

B.2 Comparison with A-Mem at Fixed K

We truncate A-Mem’s retrieved turns (including neighbor expansion) at fixed K, and compare with HiGMem’s natural output.

Method	K	Precision@K	Recall@K
A-Mem	8	0.059	0.385
A-Mem	16	0.038	0.478
A-Mem	32	0.024	0.580
A-Mem	99.84 (full)	0.010	0.750
HiGMem	8.09 (natural)	0.190	0.724

Table 8: Fixed-K retrieval comparison with A-Mem. A-Mem requires a much larger expanded context to approach HiGMem’s natural-output recall.

B.3 End-to-End Cost Analysis

To substantiate the claim that retrieving fewer turns effectively reduces downstream inference costs, we conduct an end-to-end cost analysis on the Lo-CoMo10 dataset. In the following analysis, “avg K” refers to the average number of turns ultimately provided to the answer-generation LLM per question, and “MC” denotes Memory Construction.

Token Shift. As detailed in Table 9, HiGMem spends more tokens and time during memory construction and retrieval-stage reasoning than A-Mem. However, under the hybrid deployment accounting in Table 10, this cost is shifted away from the expensive answer model: GPT-5 answer tokens fall from 25.386M to 1.629M, i.e., from roughly 12.8K to 0.8K tokens per question.

Deployment Cost Savings. Table 10 simulates a deployment in which memory construction and retrieval use GPT-4o-mini, while final answer generation uses GPT-5. Because HiGMem passes a much smaller final evidence set to the expensive

Metric	A-Mem	HiGMem
Overall F1	0.37	0.49
Avg K (Answer Stage)	99.84	8.09
MC Tokens	9.2M	21.2M
MC Time	12.8h	23.6h
QA Tokens	25.5M	29.3M
QA Time	2.4h	4.2h

Table 9: Breakdown of F1 score, retrieval size, token usage, and time consumption during the memory construction (MC) and question answering (QA) stages.

Cost Component	A-Mem	HiGMem
Mini Tokens (MC+Ret.)	11.8M	54.5M
GPT-5 Tokens (Answer)	25.4M	1.6M
Mini Cost	\$1.44	\$5.30
GPT-5 Cost	\$15.99	\$1.12
Total Cost	\$17.43	\$6.43

Table 10: Deployment cost simulation (MC and retrieval utilizing GPT-4o-mini, while answer generation utilizes GPT-5).

answer model, the estimated total cost drops from \$17.43 to \$6.43, about a $2.7\times$ reduction.

B.4 Generalization to DialSim Dataset

To evaluate the generalization capability of our approach to a second dataset, we conduct preliminary experiments on DialSim v1.1, a multi-party TV show dialogue dataset (including *Friends*, *The Big Bang Theory*, and *The Office*). We sampled 7,000 turns and 3,000 questions across three shows under a shared manifest. Note that HiGMem was not specifically tuned for this multi-party setting.

Metric	A-Mem	HiGMem
Overall F1	0.49	0.42
Avg K	95.8	3.9
MC Prompt Tokens	9.3M	25.2M
MC Time	9.1h	15.1h
QA Prompt Tokens	36.7M	23.5M
QA Time	2.9h	4.0h

Table 11: F1 score, retrieval size, and resource breakdown on the DialSim dataset (3,000 questions).

Cost Component	A-Mem	HiGMem
Mini Tokens (MC+Ret.)	12.0M	52.5M
GPT-5 Tokens (Answer)	36.5M	1.4M
Mini Cost	\$1.45	\$5.10
GPT-5 Cost	\$22.96	\$1.06
Total Cost	\$24.42	\$6.17

Table 12: Deployment cost on DialSim (GPT-4o-mini for MC/Retrieval, GPT-5 for Answer).

As shown in Table 11, A-Mem achieves a higher F1 score (0.49 vs. 0.42), which is primarily due to its much larger retrieval context (avg K=95.8 vs. 3.9). However, this does not diminish our core contribution of LLM-driven reasoning-based retrieval and memory structuring. The highly compact average K demonstrates that our retrieval paradigm remains effective at precise evidence localization. The F1 gap may reflect that the current two-level event-turn structure is not yet fully optimized for multi-party dialogue. Despite the lower F1, Table 12 shows that HiGMem’s downstream token efficiency advantage remains robust, proving to be $\sim 4.0\times$ cheaper overall in a hybrid deployment scenario. Exploring more generalizable memory structures for multi-party settings is a promising direction for future work.

B.5 Storage and Retrieval Scaling

To address potential concerns regarding storage and retrieval efficiency as the memory size grows, we analyze the scaling behaviors of both methods. Note that the retrieval time here measures pure vector-only computation (cosine top-K) and excludes LLM API calls.

turns	A-Mem	HiGMem
	(Storage / Latency)	(Storage / Latency)
100	0.20MB / 0.038ms	0.26MB / 0.042ms
10K	15.4MB / 0.15ms	16.7MB / 0.20ms
100K	153.6MB / 4.5ms	166.7MB / 4.8ms
1M	1536MB / 47.7ms	1667MB / 53.0ms

Table 13: Scaling analysis of storage footprint and vector-only retrieval latency across different memory sizes.

As illustrated in Table 13, both approaches scale linearly with the number of turns. HiGMem incurs approximately 8.5% additional storage overhead due to the required event-layer data. Furthermore, the vector-only retrieval latency remains well within a $\sim 10\%$ margin compared to A-Mem at all scales. This demonstrates that the hierarchical structure of HiGMem introduces negligible computational overhead to the underlying vector search process.

B.6 Open-Source Backbone: Qwen2.5-3B

We further evaluate HiGMem with Qwen2.5-3B on *LoCoMo10*, comparing against A-Mem under the same backbone. This experiment was run without query rewriting, so it is intended as a supplementary robustness check rather than the main reported

setting. The trend remains consistent: HiGMem improves overall F1 while retrieving substantially fewer turns.

Metric	A-Mem	HiGMem
Overall F1	0.34	0.42
Avg K	76.54	21.68
Precision@K	0.011	0.051
Recall@K	0.634	0.657

Table 14: Supplementary Qwen2.5-3B comparison on LoCoMo10.

C Prompt Summaries

We provide the complete prompt templates in the released code. In this appendix, we summarize the four prompt families most important for reproducing HiGMem on LoCoMo10: turn analysis, event affiliation, retrieval-stage filtering, and category-specific final QA generation. The summaries below describe the practical role, input structure, and output format of each prompt family, rather than reproducing the full templates verbatim.

Turn Analysis Prompt. This prompt is used when a new dialogue turn is written into memory. Its input consists of a short sliding window of recent dialogue turns together with the current turn, including speaker, timestamp, and text. The model is asked to analyze the current turn semantically. The output is a structured JSON object containing metadata (e.g., keywords, context, timestamp and tags), profile-retrieval keys, and direct links. This prompt supports the turn layer by converting raw dialogue into a structured memory unit that can later participate in retrieval and event construction.

Event Affiliation Prompt. This prompt determines whether a newly added turn should attach to one or more existing events or start a new event. Its input includes the new turn and a small set of top candidate events retrieved by similarity, together with optional recent event context. The model is asked to compare the new turn against these candidate events and output a structured affiliation decision, including selected event IDs or `NEW_EVENT`. This prompt is the main mechanism that builds the event layer online and groups related turns into higher-level memories rather than leaving the memory as a flat set of turns.

Retrieval-Stage Filtering Prompt. This prompt family is used during question answering to reduce

a broad candidate pool into the final evidence turns shown to the answer-generation model. Its input includes the user question and candidate memory items gathered from vector retrieval, event-based prediction, and optional link expansion. The model is asked to judge which candidate items are useful for answering the question, with special attention to subject matching, topical relevance, and temporal constraints. The output is a JSON object containing the IDs of candidate items judged relevant. Before this final filtering step, HiGMem also performs an event-local selection step that asks the model which turns inside a retrieved event are most likely to contain the answer. Together, these prompts turn a large candidate pool into a smaller and more precise evidence set.

Final QA Generation Prompt. For the LoCoMo10 setting used in our main HiGMem runs, final QA generation uses a concise category-specific prompt family rather than a single generic prompt. All variants take as input the retrieved turn context and the question; adversarial questions additionally include the distractor answer candidate. For temporal questions, the model is instructed to infer an approximate date from the dialogue timeline and answer briefly. For adversarial questions, the model is constrained to choose between the candidate answer and *Not mentioned in the conversation*. For the remaining categories, the model is instructed to answer with a short phrase and to use exact words from the retrieved context whenever possible. These prompts ensure short, category-appropriate, and evidence-grounded final answers.