

Calibrating Hierarchical Memory Utilization for Long-Term Memory QA

Hang Lv, Xuyang Zhi, Hongchao Gu, Hao Wang[†]

University of Science and Technology of China
{lvhang1001, zxy_zds, hcgu}@mail.ustc.edu.cn
wanghao3@ustc.edu.cn

Abstract

Long-term memory systems increasingly organize interaction histories hierarchically, with coarse summaries, mid-level events, and fine-grained evidence providing complementary signals. Yet after retrieval, these signals are often collapsed into a single generation context and used implicitly. We ask how retrieved hierarchical memory should shape generation after access, framing it as a post-retrieval calibration problem. We propose HiR, a training-free inference-time framework that contrasts full-hierarchy decoding with a coarse-only reference view, preserving the complete hierarchy while tempering over-reliance on broad abstractions. Extensive experiments and analyses show that HiR enables more effective and robust use of hierarchical memory without changing memory construction or retrieval; more broadly, our results suggest that memory research should consider not only what to store or retrieve, but also how retrieved information shapes generation. Code is available at <https://github.com/USTC-StarTeam/HiR>.

1 Introduction

Memory is essential for turning large language models from isolated prompt responders into agents that can maintain continuity across interactions (Lewis et al., 2020; Wang et al., 2023). However, useful memory is not a passive archive that simply stores past observations and appends them to the prompt; it requires accumulated information to be maintained in a form that remains accessible, compact, and relevant to future interactions (Zhong et al., 2024; Packer et al., 2024; Gutiérrez et al., 2025a; Chhikara et al., 2025). This challenge becomes especially pronounced in long-term settings, where recent benchmarks show that even strong LLMs and long-context models struggle to handle information accumulated over long, evolving, and multi-session histories (Maharana et al., 2024; Wu et al., 2025; Hu et al., 2026a).

Existing work has mainly addressed this challenge from two perspectives. One line focuses on memory construction: modern long-term memory systems organize past interactions into persistent structures, including reflection-based memories over raw experiences (Park et al., 2023; Tan et al., 2025), user profiles or preference memories (Zhong et al., 2024; Chhikara et al., 2025), tiered or archival memory stores (Packer et al., 2024), graph-structured memories (Gutiérrez et al., 2025a; Chhikara et al., 2025), and multi-granularity summaries or abstraction layers (Tan et al., 2025; Sun et al., 2026; Zhang et al., 2026; Cao et al., 2026). Another line focuses on memory access and selection: recent methods improve how systems surface query-relevant evidence through retriever or reranker optimization (Shi et al., 2024; Yu et al., 2024), adaptive retrieval decisions based on uncertainty, query complexity, or self-reflection (Jiang et al., 2023; Asai et al., 2023; Jeong et al., 2024), and routing over graph, timeline, or hierarchical memories (Gutiérrez et al., 2025b; Tan et al., 2025; Sun et al., 2026; Cao et al., 2026).

Together, these studies show that hierarchical structure supports information organization and retrieval (Zhang et al., 2024; Hu et al., 2026b; Chen et al., 2024a, 2026). However, retrieved memory components are often collapsed into a generation context and used implicitly by the generator (Hu and Lu, 2025). This raises a complementary question: should memory-augmented generation only decide what information is present, or also calibrate how strongly that information shapes the answer?

We answer this question by shifting the focus from memory access to post-retrieval influence. We propose **HiR**, an inference-time framework that leaves memory construction and retrieval unchanged, but calibrates how the retrieved hierarchy guides generation. HiR is motivated by the dual role of coarse memory: it provides useful global context, but may also act as a high-level prior that

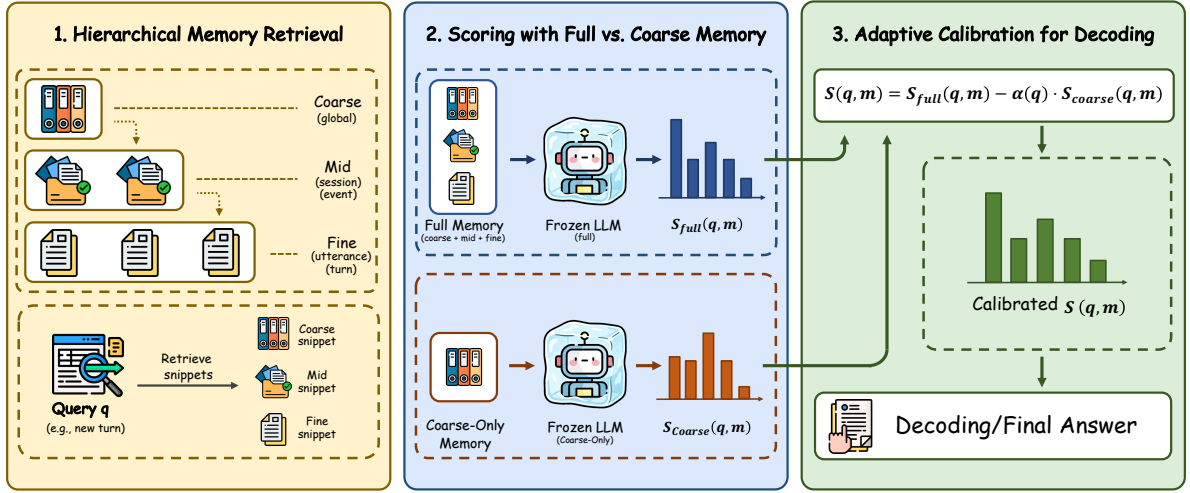


Figure 1: **Overview of HiR.** Given retrieved multi-level memory, HiR forms a full-hierarchy view and a coarse-only reference view. The same frozen answer model produces decoding signals under both views, and HiR calibrates the full-view preference by subtracting an adaptively weighted coarse-view signal. This preserves access to the complete hierarchy while controlling the influence of broad coarse-level memory during generation.

pulls the model toward broad or underspecified answers. To control this influence, HiR contrasts full-hierarchy decoding with a coarse-only reference view, tempering coarse-level signals while preserving the complete hierarchy. This yields a training-free mechanism for graded memory utilization without modifying the generation model.

Experiments across two long-term memory benchmarks and five open-source LLMs show that HiR improves over baselines in most settings, highlighting the value of post-retrieval calibration. Additional controlled analyses show that HiR remains robust across hierarchy organizations and retrieval backends, suggesting that its benefits are not tied to a specific memory construction or retrieval setting. Our contributions can be summarized as follows:

- **Post-Retrieval Memory Calibration:** We introduce a new perspective on hierarchical memory utilization, shifting memory-augmented generation from retrieval decisions to calibrating how retrieved memory influences generation.
- **The HiR Framework:** We propose HiR, a training-free inference-time framework that contrasts full-hierarchy decoding with a coarse-only reference, tempering broad memory signals while preserving the complete hierarchy.
- **Empirical Validation:** We evaluate HiR across benchmarks, LLMs, retrievers, and memory variants, showing gains over full-hierarchy prompting in most settings and providing controlled analyses of its robustness and design choices.

2 Method

2.1 Preliminaries: Hierarchical Memory

Recent long-term memory systems increasingly adopt hierarchical organizations of past interactions, maintaining information at multiple levels of abstraction (Maharana et al., 2024). Formally, let $\mathcal{D} = \{\mathcal{D}^c, \mathcal{D}^m, \mathcal{D}^f\}$ denote a hierarchical memory store, where $\mathcal{D}^c$, $\mathcal{D}^m$, and $\mathcal{D}^f$ correspond to coarse-level, mid-level, and fine-grained memories, respectively. Coarse memories capture global summaries, profiles, or high-level temporal context; mid-level memories capture sessions, events, or local episodes; and fine memories preserve raw dialogue turns or fine-grained observations. These levels trade off coverage and specificity: higher levels provide compact global context, while lower levels retain more localized evidence. Given a query q , an existing retriever or memory system surfaces a query-specific hierarchy $\mathcal{M}_q = \{\mathcal{M}_q^c, \mathcal{M}_q^m, \mathcal{M}_q^f\}$, where $\mathcal{M}_q^\ell \subseteq \mathcal{D}^\ell$. HiR takes this retrieved hierarchy as input and studies how it should influence generation, without assuming a particular memory construction or retrieval algorithm.

2.2 Full-Coarse Views

Given the retrieved hierarchy $\mathcal{M}_q$, HiR forms two memory-conditioned views. The full view exposes all retrieved levels to the answer model:

$$C_{\text{full}}(q) = [\mathcal{M}_q^c; \mathcal{M}_q^m; \mathcal{M}_q^f].$$

The coarse view keeps only the high-level memory:

$$C_{\text{coarse}}(q) = [\mathcal{M}_q^c].$$

These two contexts define the full-hierarchy and high-level-only views used below.

We use the same frozen answer model under both views. At decoding step t , conditioned on the generated prefix $y_{<t}$, the model produces two log-probability vectors over the vocabulary $\mathcal{V}$:

$$\begin{aligned}\ell_{\text{full}}^t &= \log \text{softmax}(f_\theta(q, C_{\text{full}}(q), y_{<t})), \\ \ell_{\text{coarse}}^t &= \log \text{softmax}(f_\theta(q, C_{\text{coarse}}(q), y_{<t})).\end{aligned}$$

Here, f_θ denotes the frozen answer model’s next-token scoring function, and $\ell_{\text{full}}^t(v)$ and $\ell_{\text{coarse}}^t(v)$ denote the log-probabilities assigned to token $v \in \mathcal{V}$ under the full and coarse views, respectively. The full-view distribution captures the model’s decoding preference when the hierarchy is available, while the coarse-view distribution captures the preference induced by high-level memory alone.

2.3 Calibrated Hierarchical Decoding

Given the full-view and coarse-view decoding signals, HiR calibrates the final token preference by contrasting the full-memory log-probability with a weighted coarse-memory reference:

$$S_{\text{HiR}}^t(v) = \ell_{\text{full}}^t(v) - \alpha_t \ell_{\text{coarse}}^t(v), \quad v \in \mathcal{V}.$$

Here, $S_{\text{HiR}}^t(v)$ is a calibrated decoding score rather than a normalized log-probability. The coarse-view log-probability is used only as a reference term in this score, estimating the token preference induced by high-level memory alone.

The coefficient α_t is adapted at each decoding step according to the entropy gap between the two memory-conditioned distributions:

$$\begin{aligned}\Delta H_t &= H(p_{\text{coarse}}^t) - H(p_{\text{full}}^t), \\ \alpha_t &= g_{\alpha_0}(\Delta H_t).\end{aligned}$$

Here, $p_{\text{full}}^t = \exp(\ell_{\text{full}}^t)$, $p_{\text{coarse}}^t = \exp(\ell_{\text{coarse}}^t)$, $H(p) = -\sum_{v \in \mathcal{V}} p(v) \log p(v)$, and g_{α_0} is a bounded monotonic mapping with base strength α_0 . This step-wise coefficient lets HiR adjust the calibration strength locally during decoding rather than using a fixed global penalty. When the full hierarchy yields a sharper distribution than the coarse view, ΔH_t becomes positive and HiR applies a stronger correction against coarse-only preferences. When the full view is less decisive, the correction is reduced. The calibrated scores S_{HiR}^t replace the original full-view log-probabilities in standard decoding, requiring no training and no modification to the underlying generation model.

Dataset	Model	NoMem	Flat	Full	HiR
LoCoMo	Qwen3.5-4B	8.9	34.1	35.6	40.8
	Qwen3.5-9B	9.2	35.9	39.8	45.1
	Qwen3.5-27B	6.8	31.9	34.5	41.3
	Qwen3-8B	12.9	41.9	45.6	47.8
	Llama3.1-8B	4.2	38.8	45.6	46.4
LongMem	Qwen3.5-4B	8.0	47.8	47.2	48.2
	Qwen3.5-9B	9.8	50.6	50.0	51.2
	Qwen3.5-27B	11.6	58.4	60.2	59.4
	Qwen3-8B	11.8	47.0	46.4	50.4
	Llama3.1-8B	8.2	48.6	48.0	49.6

Table 1: Main results on LoCoMo and LongMemEval-s, reported as percentages, where **bold** values indicate the best result in each row and FULL denotes the full hierarchy without HiR calibration.

3 Experiments

3.1 Experimental Setup

Benchmarks. We evaluate HiR on LoCoMo (Maharana et al., 2024) and the LongMemEval-s split (Wu et al., 2025), two long-term memory QA benchmarks. For LoCoMo, we follow the open-ended QA protocol and report token-level F1. For LongMemEval-s, we follow the official LLM-judge prompt and report judge accuracy using GPT-5-mini as the evaluator. For both benchmarks, we organize retrieved evidence into coarse, mid, and fine memory contexts; dataset-specific construction details are provided in Appendix A.

Models. We conduct the experiments on five open-source instruction-tuned LLMs, including: Qwen3.5-4B, Qwen3.5-9B, Qwen3.5-27B (Qwen Team, 2026), Qwen3-8B (Yang et al., 2025), and Llama3.1-8B-Instruct (Grattafiori et al., 2024). These models cover different sizes and families.

Baselines. We compare HiR with three memory-use baselines: no_memory, which answers without retrieved memory; flat_raw, which uses raw evidence without explicit hierarchy and tree_full, which exposes the full coarse/mid/fine hierarchy without calibration. HiR uses the same full hierarchy as tree_full, but applies full-vs-coarse calibration during decoding. We further analyze level-only and level-combination variants.

3.2 Main Results

HiR improves full-hierarchy prompting. Table 1 shows that HiR improves over the uncalibrated FULL baseline on all five LoCoMo models, with an average gain of 4.1 F1 points. The gains span different Qwen sizes, Qwen generations, and Llama3.1, suggesting that HiR is not tied to a single architec-

Model	Coarse	Mid	Fine	M+F	Full	HiR
Qwen3.5-4B	7.7	24.2	34.9	37.2	35.6	40.8
Qwen3.5-9B	7.8	25.7	36.9	39.9	39.8	45.1
Qwen3.5-27B	7.5	22.9	34.0	36.3	34.5	41.3
Qwen3-8B	10.0	30.1	40.6	44.7	45.6	47.8
Llama3.1-8B	9.3	31.3	39.1	45.0	45.6	46.4

Table 2: LoCoMo level-context analysis. Scores are percentages; best results are **bolded**.

ture or scale. Qwen3.5-27B has a relatively low absolute LoCoMo F1 because it often appends short explanatory phrases, reducing lexical overlap with the reference despite preserving the intended answer. On LongMemEval-s, HiR achieves the best result on four of five models and improves over FULL by 1.4 points on average, indicating generally positive but more model-sensitive behavior.

Coarse memory is useful but needs calibration. Table 2 examines how different memory granularities contribute to LoCoMo performance. Single-level contexts are consistently weaker than FULL and HiR, showing that hierarchical memory provides complementary signals across levels. Removing coarse memory with M+F sometimes improves over FULL, suggesting that uncalibrated coarse context can interfere, but it never matches HiR. Moreover, FULL outperforms M+F for Qwen3-8B and Llama3.1-8B, indicating that coarse memory also provides useful global signal. Thus, HiR improves not by discarding coarse memory, but by preserving the full hierarchy while calibrating coarse-level influence during decoding.

3.3 Discussion and Analysis

To understand when and why HiR works, we run controlled analyses on Qwen3.5-9B and LoCoMo from three perspectives: penalty sensitivity, retriever/hierarchy robustness, and alternative level reweighting. Detailed experiment settings and results are provided in Appendix B.

Adaptive calibration avoids manual penalty tuning. We first examine how fixed coarse-penalty strength affects calibration. As shown in Figure 2, fixed penalties exhibit a clear rise-and-fall pattern: moderate penalties improve over FULL, while overly large penalties suppress useful global context and sharply degrade performance. This trend indicates that coarse-memory calibration is beneficial, but the penalty must be chosen carefully. We find that the adaptive variant matches the best performance without manually searching over α , serving as a practical tuning-free default.

HiR remains robust across retrieval and hierarchy choices. We next examine whether the gain persists under different retrieval backends and hierarchy interfaces. Replacing the default lexical retriever with BM25, BGE-M3 dense retrieval (Chen et al., 2024b), or BM25 retrieval followed by BGE-reranker-v2-m3 reranking improves absolute scores, but HiR remains above the matched FULL baseline in every retriever setting (Appendix B.2; Table 4). We also test a LoCoMo-native artifact hierarchy instead of the standard hierarchy used in the main table, and observe the same qualitative trend: HiR improves over the matched native FULL baseline (Appendix B.3; Table 5). These checks suggest that HiR acts as a post-retrieval calibration method rather than an artifact of a particular retriever or hierarchy interface.

What makes the coarse reference useful? Finally, we test three alternative explanations for HiR’s gain (Appendix B.4; Table 6). (i) *Does level-aware suppression help beyond coarse memory?* Penalizing mid-level memory or jointly penalizing mid and coarse memory also improves over FULL, confirming the value of level-aware calibration. However, both remain below HiR, suggesting that coarse-reference calibration is the strongest instantiation. (ii) *Is HiR just increasing reliance on fine evidence?* Directly boosting fine memory gives almost no gain, and naively mixing coarse, mid, and fine component scores performs poorly. (iii) *Is coarse memory merely a bad prior?* Replacing the coarse reference with a no-memory prior is also weaker, showing that the benefit does not come from subtracting an arbitrary background distribution. Together, these results suggest that coarse memory is a meaningful high-level reference: useful enough to keep, but important to calibrate.

4 Conclusion

We introduced HiR, a training-free approach for post-retrieval calibration of hierarchical memory. By contrasting full-hierarchy decoding with a coarse-only reference view, HiR preserves the complete memory context while explicitly controlling broad high-level influence. Experiments and controlled analyses show that HiR improves hierarchical memory utilization and remains robust across key retrieval, hierarchy, and variant choices. These results highlight graded memory influence as a useful and complementary perspective for long-term memory-augmented generation.

Limitations

Our work has three main scope limitations. First, part of our central contribution is a new perspective on hierarchical memory utilization: retrieved memory should be calibrated in how it influences generation, rather than only selected or discarded. HiR operationalizes this perspective through a simple decoding-time full-vs-coarse calibration, but this is only one possible realization. Other designs, such as learned level controllers, training-time memory-use objectives, or retriever-side influence estimation, may further expand this direction. Second, HiR assumes that a multi-level memory interface is already available. This assumption is aligned with recent long-term memory systems, where interaction histories are commonly organized as raw evidence, session/event summaries, profiles, timelines, or other structured abstractions. Our work treats such memory structures as the input interface and focuses on their generation-time utilization. Memory ingestion, summarization, updating, and retrieval optimization are important but orthogonal problems outside the scope of this paper. Third, HiR requires an additional coarse-view forward pass during decoding. This increases inference cost compared with direct full-context prompting and requires access to token-level probabilities, which may limit direct deployment in black-box API settings.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Nos. U23A20319, 62441227, and 62472394), as well as the Anhui Province Science and Technology Innovation Project (Nos. 202423k09020010 and 202423k09020011).

References

- Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. [Self-rag: Learning to retrieve, generate, and critique through self-reflection](#). *Preprint*, arXiv:2310.11511.
- Shuqi Cao, Jingyi He, and Fei Tan. 2026. [Higmem: A hierarchical and llm-guided memory system for long-term conversational agents](#). *Preprint*, arXiv:2604.18349.
- Huiyao Chen, Yi Yang, Yinghui Li, Meishan Zhang, Baotian Hu, and Min Zhang. 2026. [Beyond chunking: Discourse-aware hierarchical retrieval for long document question answering](#). In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18176–18198, San Diego, California, United States. Association for Computational Linguistics.
- Huiyao Chen, Yu Zhao, Zulong Chen, Mengjia Wang, Liangyue Li, Meishan Zhang, and Min Zhang. 2024a. [Retrieval-style in-context learning for few-shot hierarchical text classification](#). *Transactions of the Association for Computational Linguistics*, 12:1214–1231.
- Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024b. [Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation](#). *Preprint*, arXiv:2402.03216.
- Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. 2025. [Mem0: Building production-ready ai agents with scalable long-term memory](#). *Preprint*, arXiv:2504.19413.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. [The llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.
- Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. 2025a. [Hipporag: Neurobiologically inspired long-term memory for large language models](#). *Preprint*, arXiv:2405.14831.
- Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. 2025b. [From rag to memory: Non-parametric continual learning for large language models](#). *Preprint*, arXiv:2502.14802.
- Yuanzhe Hu, Yu Wang, and Julian McAuley. 2026a. [Evaluating memory in llm agents via incremental multi-turn interactions](#). *Preprint*, arXiv:2507.05257.
- Yucheng Hu and Yuxing Lu. 2025. [Rag and rau: A survey on retrieval-augmented language model in natural language processing](#). *Preprint*, arXiv:2404.19543.
- Yuyang Hu, Shichun Liu, Yanwei Yue, Guibin Zhang, Boyang Liu, Fangyi Zhu, Jiahang Lin, Honglin Guo, Shihan Dou, Zhiheng Xi, Senjie Jin, Jiejun Tan, Yanbin Yin, Jiongnan Liu, Zeyu Zhang, Zhongxiang Sun, Yutao Zhu, Hao Sun, Boci Peng, and 28 others. 2026b. [Memory in the age of ai agents](#). *Preprint*, arXiv:2512.13564.
- Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong Park. 2024. [Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7036–7050, Mexico City, Mexico. Association for Computational Linguistics.

- Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. [Active retrieval augmented generation](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992, Singapore. Association for Computational Linguistics.
- 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. In *Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS '20*, Red Hook, NY, USA. Curran Associates Inc.
- Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 2024. [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)*, pages 13851–13870, Bangkok, Thailand. Association for Computational Linguistics.
- Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. 2024. [Memgpt: Towards llms as operating systems](#). *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, UIST '23*, New York, NY, USA. Association for Computing Machinery.
- Qwen Team. 2026. [Qwen3.5: Towards native multi-modal agents](#).
- Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Richard James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2024. [REPLUG: Retrieval-augmented black-box language models](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8371–8384, Mexico City, Mexico. Association for Computational Linguistics.
- Haoran Sun, Shaoning Zeng, and Bob Zhang. 2026. [H-MEM: Hierarchical memory for high-efficiency long-term reasoning in LLM agents](#). In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 341–350, Rabat, Morocco. Association for Computational Linguistics.
- Zhen Tan, Jun Yan, I-Hung Hsu, Rujun Han, Zifeng Wang, Long Le, Yiwen Song, Yanfei Chen, Hamid Palangi, George Lee, Anand Rajan Iyer, Tianlong Chen, Huan Liu, Chen-Yu Lee, and Tomas Pfister. 2025. [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)*, pages 8416–8439, Vienna, Austria. Association for Computational Linguistics.
- Weizhi Wang, Li Dong, Hao Cheng, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. 2023. [Augmenting language models with long-term memory](#). *Preprint*, arXiv:2306.07174.
- Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. 2025. [Longmemeval: Benchmarking chat assistants on long-term interactive memory](#). *Preprint*, arXiv:2410.10813.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.
- Yue Yu, Wei Ping, Zihan Liu, Boxin Wang, Jiaxuan You, Chao Zhang, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Rankrag: unifying context ranking with retrieval-augmented generation in llms. In *Proceedings of the 38th International Conference on Neural Information Processing Systems, NIPS '24*, Red Hook, NY, USA. Curran Associates Inc.
- Ningning Zhang, Xingxing Yang, Zhizhong Tan, Weiping Deng, and Wenyong Wang. 2026. [Himem: Hierarchical long-term memory for llm long-horizon agents](#). *Preprint*, arXiv:2601.06377.
- Zeyu Zhang, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Quanyu Dai, Jieming Zhu, Zhenhua Dong, and Jirong Wen. 2024. [A survey on the memory mechanism of large language model based agents](#). *Preprint*, arXiv:2404.13501.
- 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 Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence, AAAI'24/IAAI'24/EAAI'24*. AAAI Press.

A Hierarchical Memory Construction

Overview. Hierarchical or multi-granularity organization has become a common interface for long-term memory systems, where the same interaction history may be represented simultaneously as raw turns, local events, session summaries, timelines, user profiles, or higher-level abstractions. Our experiments adopt this interface rather than proposing a new memory construction algorithm. Specifically, we map the memory artifacts available in each benchmark into a shared coarse/mid/fine format: coarse memory provides broad temporal or session-level background, mid-level memory captures localized events or session summaries, and fine memory retains directly grounded evidence such as dialogue turns, observations, or raw facts. This unified hierarchy allows us to study HiR’s target question consistently across datasets: once multi-granularity memory has been surfaced to the generator, how should its influence on generation be calibrated?

LoCoMo. LoCoMo contains long conversational histories together with structured artifacts such as dialogue turns, observations, and session- or event-level summaries. We use these native artifacts to organize memory into the three levels used in our experiments. The coarse level contains broad session-level or timeline-level context; the mid level contains localized session or event summaries; and the fine level contains raw dialogue turns, observations, and other fine-grained evidence. Importantly, LoCoMo does not prescribe a single official coarse/mid/fine retrieval tree. Our hierarchy is therefore an experimental organization of the dataset’s native artifacts, designed to provide a unified interface for studying hierarchical memory utilization.

LongMemEval-s. LongMemEval-s does not provide the same explicit three-level memory artifacts as LoCoMo. To enable comparable evaluation, we construct an analogous hierarchy from its available session, round, turn, and timestamp structure. The coarse level represents broad temporal or session-level context, the mid level groups evidence at the session or event scale, and the fine level consists of raw turns or other directly relevant textual evidence. This construction is intended to match the same coarse/mid/fine interface used for LoCoMo, rather than to introduce a new dataset-specific memory algorithm.

Retrieval and organization. Unless otherwise specified, our main experiments use a default lexical retriever to select query-relevant memory evidence. We then organize the retrieved evidence into the coarse, mid, and fine memory contexts described above. This retrieval setup has two separate components: the retriever used to score evidence, and the search scope used for fine-grained evidence. In the main setting, we use `leaf_session_mode=global`: fine-grained evidence is retrieved from the global pool of available turns, observations, or raw facts, and the selected items are then grouped back into the corresponding session or event structure. This differs from a top-down-style setting, where the system first selects a small set of mid-level sessions and searches for fine-grained evidence only inside those selected sessions. We study alternative retrievers, including BM25, dense retrieval, and reranking variants, in Appendix B.2.

Pilot comparison of leaf retrieval scopes. We conducted an early LoCoMo pilot to compare the two fine-grained retrieval scopes. In the *selected* mode, fine evidence is retrieved only within the sessions selected by mid-level retrieval, approximating a top-down search. In the *global* mode, fine evidence is retrieved globally from observations and dialogue turns, and the retrieved items are then grouped back into the hierarchy. This pilot used an MC-style accuracy metric rather than the official QA F1 used in our main experiments.

Model	Selected leaf	Global leaf	Δ
Qwen2.5-1.5B	0.8233	0.8533	+0.0300
Qwen2.5-7B	0.8267	0.8783	+0.0516
Qwen3-1.7B	0.8683	0.9067	+0.0384
Qwen3-8B	0.8467	0.8950	+0.0483

Table 3: Pilot comparison of selected versus global leaf retrieval scopes on LoCoMo. Scores are MC-style accuracy from an early pilot and are used only to justify the default retrieval scope.

These results motivated our default `leaf_session_mode=global` setting: retrieving fine-grained evidence globally was more robust than first narrowing the search space by mid-level sessions.

Prompting across baselines. All baselines use the same answer model and the same base prompt format whenever possible. They differ primarily in the memory context exposed to the

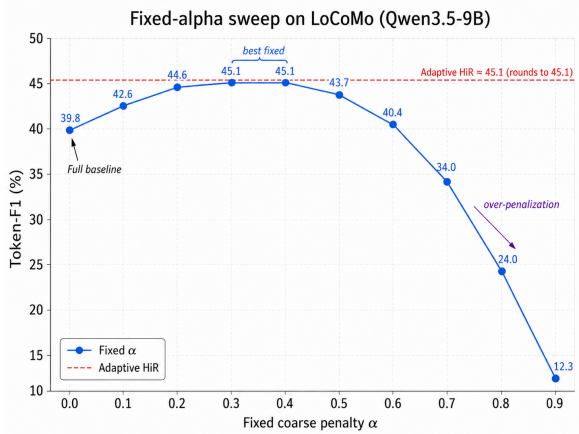


Figure 2: Fixed-alpha sweep on LoCoMo with Qwen3.5-9B. The solid curve shows fixed coarse penalties, while the dashed line shows adaptive HiR. Moderate fixed penalties improve over FULL, but overly large penalties sharply degrade performance. Adaptive HiR matches the best rounded fixed-alpha performance while avoiding manual coefficient tuning.

model. `no_memory` provides no retrieved memory. `flat_raw` provides retrieved raw or fine-grained evidence without an explicit hierarchy. `tree_fine` provides only fine-level memory. `tree_full` provides the full coarse/mid/fine hierarchy. HiR uses the same full hierarchical context as `tree_full`, but additionally runs a coarse-only reference view during decoding to calibrate the influence of high-level memory.

B Additional Controlled Analyses

This section provides the controlled analyses discussed in Section 3.3. Unless otherwise stated, all experiments use Qwen3.5-9B on LoCoMo under the official open-ended QA protocol with token-level F1. We keep the answer model, prompt format, generation setting, and evaluation protocol fixed, and vary one factor at a time: the penalty coefficient, retrieval backend, hierarchy interface, or calibration variant.

B.1 Fixed-Alpha Sweep

We first study how sensitive coarse-reference calibration is to the penalty strength. For a fixed coefficient α , we decode with

$$S^t(v) = \ell_{\text{full}}^t(v) - \alpha \ell_{\text{coarse}}^t(v),$$

where $\alpha = 0$ corresponds to the uncalibrated FULL baseline. We sweep α from 0.0 to 0.9 and compare the resulting curve with the adaptive HiR variant.

Retriever	Full	HiR	Δ
Default	39.8	45.1	+5.2
BM25	41.4	45.2	+3.9
BGE-M3 Dense	41.7	47.5	+5.8
BM25+BGE-reranker	44.0	48.8	+4.8

Table 4: Retriever robustness on LoCoMo with Qwen3.5-9B. Scores are token-F1 percentages, and Δ denotes HiR minus the matched FULL baseline under the same retriever. BGE-M3 dense uses the BGE-M3 embedding model, while BM25+BGE-reranker first retrieves candidates with BM25 and then reranks them with BGE-reranker-v2-m3.

Figure 2 shows a clear rise-and-fall pattern. Moderate fixed penalties improve over FULL, confirming that calibrating coarse-level influence is beneficial. However, large penalties substantially reduce performance, indicating that coarse memory also carries useful global context and should not be over-suppressed. Adaptive HiR reaches the best rounded performance without manually selecting α , making it a practical tuning-free default.

B.2 Retriever Robustness

We next evaluate whether HiR remains effective when the retrieval backend changes. Starting from the default lexical retriever, we compare BM25, BGE-M3 dense retrieval (Chen et al., 2024b), and BM25 retrieval followed by BGE-reranker-v2-m3 reranking. For each retriever, we report the matched FULL baseline and HiR under the same hierarchy, prompt, and decoding setup; only the retrieval backend is changed.

Table 4 shows that stronger retrievers improve absolute performance for both FULL and HiR. Importantly, HiR remains above the matched FULL baseline under every retriever. This supports the view that HiR improves post-retrieval memory utilization rather than compensating for a weak or idiosyncratic retriever.

B.3 LoCoMo-Native Hierarchy

We further evaluate whether HiR depends on the standard hierarchy organization used in the main experiments. To do so, we construct an alternative hierarchy directly from LoCoMo’s native structured artifacts and evaluate HiR under this LoCoMo-native hierarchy. This setting uses a different hierarchy interface from the main table, so the key comparison is matched within the same interface: NATIVE FULL versus NATIVE HiR.

Table 5 shows that HiR also improves over the

Method	Score	Δ
Native Coarse	23.8	-13.5
Native Mid	29.6	-7.6
Native Fine	27.7	-9.6
Native Full	37.2	+0.0
Native HiR	38.3	+1.1

Table 5: LoCoMo-native hierarchy results with Qwen3.5-9B. Scores are token-F1 percentages, and Δ is relative to the matched NATIVE FULL baseline.

matched NATIVE FULL baseline. This result indicates that the benefit of HiR is not tied to the standard heuristic hierarchy used in the main experiments. Rather, HiR operates on the surfaced multi-level memory interface and calibrates how that hierarchy influences generation. Together with the retriever analysis above, this supports the claim that HiR is robust to both retrieval and hierarchy choices.

B.4 Alternative Calibration Variants

Finally, we compare HiR with representative calibration and reweighting variants to clarify what role the coarse reference plays. All experiments in this subsection use the standard heuristic hierarchy with `leaf_session_mode=global`. The retriever, prompt format, hierarchy construction, answer model, and decoding setup are kept identical to the main LoCoMo experiments, so the comparison isolates the effect of changing the decoding-time calibration rule.

At decoding step t , let ℓ_{full}^t , ℓ_c^t , ℓ_m^t , ℓ_f^t , and $\ell_{\emptyset}^t$ denote the log-probability vectors produced by the same frozen answer model under the full hierarchy, coarse-only, mid-only, fine-only, and no-memory views, respectively. The uncalibrated FULL baseline uses

$$S_{\text{Full}}^t(v) = \ell_{\text{full}}^t(v).$$

HiR uses the coarse-only view as a reference:

$$S_{\text{HiR}}^t(v) = \ell_{\text{full}}^t(v) - \alpha_t^{(c)} \ell_c^t(v),$$

where $\alpha_t^{(c)}$ is the step-wise adaptive coefficient defined in Section 2.3. For variants that replace the reference view, we use the analogous notation $\alpha_t^{(r)} = g_{\alpha_0}(H(p_r^t) - H(p_{\text{full}}^t))$, where $r \in \{m, \emptyset\}$.

We organize the alternatives around three questions. First, does level-aware suppression help beyond coarse memory? We test mid-level penalties and joint mid+coarse penalties. Second, is HiR simply increasing reliance on fine evidence? We test

direct fine boosting and direct coarse/mid/fine component mixing. Third, is coarse memory merely a bad prior? We test replacing the coarse reference with a no-memory reference. For each variant family, we report a representative or best-performing configuration rather than all swept hyperparameters.

Table 6 shows that several level-aware variants improve over FULL, confirming that calibration of memory influence is useful. However, the strongest performance comes from using the coarse view as the reference. Fine boosting gives little improvement, and direct coarse/mid/fine component mixing performs poorly, indicating that HiR is not simply increasing reliance on fine-grained evidence. Replacing the coarse reference with a no-memory prior is also weaker, showing that the gain does not come from subtracting an arbitrary background distribution. These results support the interpretation that coarse memory provides a meaningful high-level reference: it is useful enough to keep, but important to calibrate.

C Potential Risks

This work studies how language models use long-term and hierarchical memory during generation. Although our experiments are conducted on public benchmarks, similar techniques may be applied to systems that store user-specific or sensitive information. Such applications introduce potential privacy risks if memory is collected, retained, or retrieved without appropriate consent and access control. In addition, improving memory utilization may also amplify incorrect, outdated, or biased memories if the underlying memory store is not properly maintained. We therefore recommend that practical deployments of long-term memory systems include explicit user consent, data minimization, memory editing and deletion mechanisms, and safeguards for auditing retrieved evidence before final generation.

D AI Usage

We used LLMs only for language polishing and assisting with parts of code implementation. The core research idea, experimental design, paper structure, and initial manuscript were developed by the human authors. All experimental results were produced and verified by the authors, and the authors take full responsibility for the claims, analyses, and conclusions presented in this paper.

Variant	Decoding rule	Score	Δ	Interpretation
FULL	$S^t(v) = \ell_{\text{full}}^t(v)$	39.8	+0.0	Full hierarchy without calibration
Adaptive Fine boost	$S^t(v) = \ell_{\text{full}}^t(v) + \alpha \ell_f^t(v)$	39.9	+0.1	Directly emphasizing fine evidence gives little gain
Adaptive mid penalty	$S^t(v) = \ell_{\text{full}}^t(v) - \alpha_t^{(m)} \ell_m^t(v)$	42.0	+2.2	Penalizing mid-level memory helps less than HiR
Adaptive mid+coarse penalty	$S^t(v) = \ell_{\text{full}}^t(v) - \alpha \ell_m^t(v) - \beta \ell_c^t(v)$	44.0	+4.2	Jointly suppressing upper levels helps, but remains below HiR
Uniform CMF	$S^t(v) = \frac{1}{3} (\ell_c^t(v) + \ell_m^t(v) + \ell_f^t(v))$	21.8	-18.0	Direct component mixing is unstable
Adaptive no-memory penalty	$S^t(v) = \ell_{\text{full}}^t(v) - \alpha_t^{(\emptyset)} \ell_{\emptyset}^t(v)$	41.9	+2.1	Generic prior subtraction is weaker than using coarse memory
Adaptive HiR	$S^t(v) = \ell_{\text{full}}^t(v) - \alpha_t^{(c)} \ell_c^t(v)$	45.1	+5.2	Coarse-reference calibration

Table 6: Representative alternative calibration variants on LoCoMo with Qwen3.5-9B. Scores are token-F1 percentages. Displayed scores are rounded to one decimal, and Δ is computed from unrounded scores relative to FULL. For variants labeled “Best”, α or (α, β) denotes the best-performing fixed coefficient(s) found in the corresponding sweep.

E Reproducibility Statement

Datasets. Our experiments use two long-term memory QA benchmarks. For LoCoMo, we evaluate on 1,542 open-ended QA examples. For LongMemEval-s, we evaluate on the 500-example split following the official LLM-judge protocol. Dataset-specific hierarchy construction details are provided in Appendix A.

Computational resources. All experiments were conducted on a local workstation equipped with two NVIDIA RTX 5090 GPUs. Including preliminary exploration, ablation studies, reruns, and the final experiments reported in the paper, the full experimental process took approximately four weeks on this two-GPU setup.

Determinism. To improve reproducibility, we set the temperature to 0 during both answer generation and LLM-based evaluation. Thus, all open-ended generations and judge calls use deterministic decoding whenever supported by the underlying model or API. We also use fixed prompts, fixed retrieved memory contexts, and fixed evaluation splits for all reported comparisons.

API-based evaluation. Since LongMemEval-s relies on API-based LLM judging, we additionally evaluated the same predictions with Qwen3.5-Plus using the same judge prompt and temperature-0 setting. The scores and trends are similar to the GPT-5-mini results reported in the main table, suggesting that our conclusions are not tied to a single evaluator model.