QueryLink: Leveraging Query-Memory Alignment for Long-Term Reasoning in LLM Agents

Xuxian Hu[1], **Zhu Teng**[1*], **Wei Zhang**[1], **Ming He**[2], **Jianping Fan**[2]
[1]Beijing Jiaotong University, [2]AI Lab at Lenovo Research
Correspondence: zteng@bjtu.edu.cn

Abstract

Retrieval-Augmented Generation (RAG) systems are widely used to mitigate the stateless nature of Large Language Models (LLMs) in long-term and personalized interactions by incorporating external memory. However, existing approaches often prioritize memory organization, such as knowledge graphs, while overlooking a critical semantic gap between implicit, intent-driven queries and explicit, narrative-based memories. To bridge this gap, we propose QueryLink, a novel framework that leverages Query-Memory Alignment to project both queries and memories into a shared semantic space. It significantly boosts recall by facilitating multi-grained retrieval of semantically relevant information. To further enhance memory retrieval, we leverage Coherent Memory Chunking, a mechanism that processes memories in multi-turn dialogue units, preserving semantic integrity, rather than relying on fixed-size segments. Extensive experiments on the LoCoMo and LongMemEval benchmark demonstrate that QueryLink significantly outperforms SOTA methods, achieving at least a 7% improvement in reasoning accuracy (measured by LLM). Additionally, QueryLink can be integrated as a plug-and-play component to boost existing vector-based systems like A-MEM, leading to improvements of over 6% in both F1 and B1 scores. The code is available at https://github.com/Dontplay0112/querylink.

1 Introduction

The rapid evolution of Large Language Models (LLMs) has enabled the development of autonomous agents capable of performing complex tasks (Lee et al., 2023; Xi et al., 2025; Xiong et al., 2026). For these agents to maintain coherence across long-term interactions, which may span days or even months, Long-Term Memory (LTM) systems are crucial(Jiang et al., 2024). Currently, Retrieval-Augmented Generation (RAG), based on vector databases, is the most widely adopted solution for integrating external memory(Gutiérrez et al., 2025). However, in practice, dense retrieval systems often suffer from retrieval failure or misaligned retrieval, where the retrieved memories do not align well with the user's current query.

We argue that this issue stems from a fundamental heterogeneity between the query space and the memory space. It mirrors the cognitive distinction between Episodic Memory (which records specific events) and Semantic Memory (which stores general knowledge)(De Brigard et al., 2022). On the Query Side, queries tend to be implicit, interrogative, and high-level (e.g., "What did we decide about the project?"), often containing missing information and relying on high-level intent. In contrast, on the Memory Side, stored memories are declarative and narrative, rich in specific entities, temporal markers, and detailed context (e.g., "User confirmed on Tuesday that API latency is the bottleneck"). Standard Embedding model, commonly used for retrieval tasks, are not trained to bridge this "Abstract-Concrete" mapping between the implicit, intent-driven nature of queries and the explicit, detail-rich nature of memories. As a result, significant semantic distance arises between related queries and memories in the embedding space, leading to retrieval failures and a decline in long-term reasoning accuracy.

Existing research has predominantly focused on optimizing Memory Construction—organizing information into hierarchies or knowledge graphs (Salama et al., 2025; Sun and Zeng, 2025; Chhikara et al., 2025; Zhang et al., 2025a)—to better represent relationships. However, these architectural advances often neglect the underlying issue of semantic alignment within the raw textual data itself. To quantify the impact of the semantic gap between the query space and the memory space, we ana-

*Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2026, pages 15608–15621

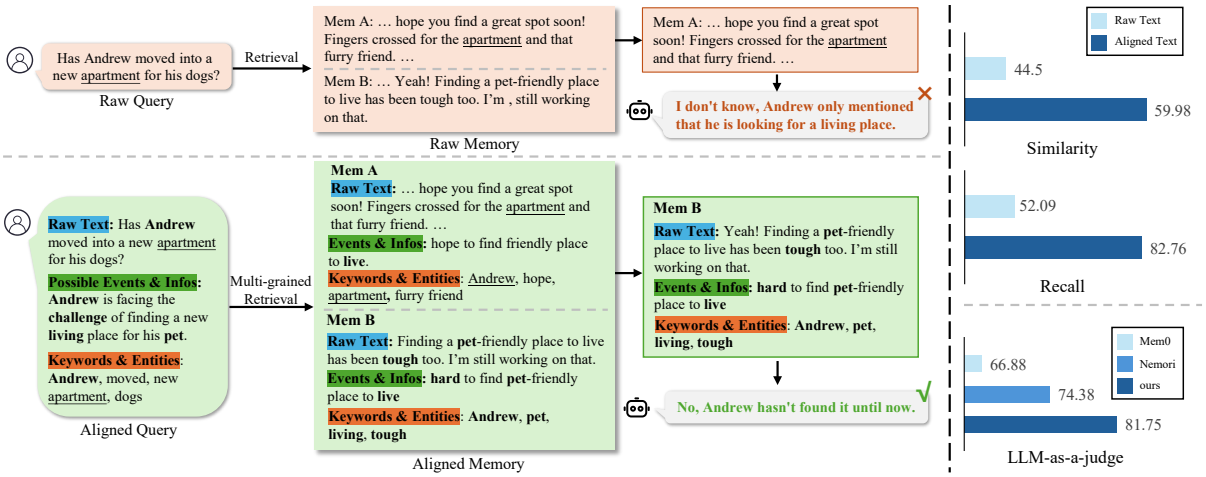

Figure 1: Conventional retrieval vs. QueryLink. Top-Left: Conventional retrieval often relies on surface-level keyword overlap (e.g., "apartment"), leading to misaligned context and incorrect answers. Bottom-Left: QueryLink designs a query-memory alignment to bridge this semantic gap, accurately retrieving the context relevant to the user's challenge in finding a "pet-friendly" apartment. Right: Experiments demonstrates that QueryLink significantly improves memory recall by enhancing semantic similarity, leading to superior performance on this task.

lyze the vector distributions of ground-truth query-memory pairs. As shown in Figure 1, the average cosine similarity between raw queries and raw memories is surprisingly low, at 0.445, which explains the frequent failures observed in naive RAG systems. However, when applying our proposed Query-Memory Alignment, the similarity increases significantly to 0.599 (a 15.4% improvement). This result suggests that in high-dimensional space, semantics should not be represented by a single point but rather anchored by multi-grained semantics, reflecting the rich and diverse nature of both queries and memories.

Based on this observation, we propose QueryLink, a framework grounded in the philosophy that effective retrieval requires establishing symmetric, multi-grained semantic representations across both the Query Side and the Memory Side. Our contributions are as follows:

- We propose a novel framework, QueryLink, for long-term memory in LLM agents. It integrates three key components: Coherent Memory Chunking, Query-Memory Alignment, and Multi-Grained Retrieval. QueryLink enhances recall by capturing both precise lexical matches and deeper semantic intents, particularly in multi-turn interactions, enabling more accurate and context-aware retrieval.
- We develop a novel Query-Memory Alignment mechanism that optimizes the matching between queries and memories by projecting both into a shared, symmetric semantic space. This alignment enhances retrieval accuracy by capturing semantic nuances at multiple granularities, allowing the system to bridge the gap between the high-level, intent-driven nature of queries and the detailed, entity-rich structure of memories.
- We conduct extensive experiments on the LoCoMo and LongMemEval benchmark, demonstrating that QueryLink sets a new state-of-the-art in agent memory, outperforming current SOTA methods by at least 7% in reasoning accuracy (measured by Judge). Furthermore, QueryLink can be seamlessly integrated as a plug-and-play component into existing systems, yielding substantial performance gains without requiring architectural redesigns, highlighting its versatility.

2 Related Work

2.1 Long-Context Interaction & External Memory

Despite the growing context windows of modern LLMs (Xi et al., 2025; Liu et al., 2024), relying solely on physical context for interactions spanning weeks or months remains impractical due to prohibitive inference costs and the well-documented "Lost-in-the-Middle" phenomenon (Liu et al., 2024). As a result, constructing Long-Term Memory (LTM) through external vector databases has become the dominant approach (Wu et al., 2025; Zhong et al., 2024). Early works, such as MemoryBank (Zhong et al., 2024) and Genera-

tive Agents (Park et al., 2023), try to enhance agent consistency by introducing mechanisms like "Memory Streams" and iterative "Reflection". However, it assume that simple cosine similarity within a pre-trained embedding space is sufficient for effective retrieval. In contrast, QueryLink argues that, in long-term interactions, there exists a significant mismatch between the abstract, intent-driven nature of user queries and the concrete, detailed nature of historical memories—a gap that raw embeddings alone cannot bridge effectively.

2.2 Semantic Alignment in RAG

To improve retrieval relevance, existing RAG research has primarily focused on Asymmetric Expansion strategies. On the query side, methods like HyDE (Gao et al., 2023a) and Query2Doc (Wang et al., 2023) project queries into the document space by generating hypothetical documents. While effective for factual QA tasks, these approaches introduce the risk of "hallucinations", where inaccurate details can mislead the retrieval process(Zhang et al., 2025b). On the memory side, approaches like MemInsight (Salama et al., 2025) summarize historical data to better match query semantics. However, these unilateral strategies are limited because they fail to establish a truly shared semantic manifold between queries and memories, leading to a mismatch in retrieval accuracy. In contrast, QueryLink addresses this gap by introducing a novel Query-Memory Alignment mechanism. This mechanism simultaneously constructs multi-grained semantic representations on both the query and memory sides, ensuring a more comprehensive and nuanced semantic matching. By leveraging multi-grained semantic alignment, QueryLink reduces the reliance on any single perspective, minimizing the risk of hallucinations and enabling more accurate, contextually aware retrieval.

2.3 Memory Organization for Agents

In pursuit of higher retrieval precision, recent studies have focused heavily on designing complex memory structures. Systems like Mem0 (Chhikara et al., 2025) and Hi-Mem (Sun and Zeng, 2025) utilize hierarchical structures or Knowledge Graphs to manage memory, while Nemori (Nan et al., 2025) leverages hypergraphs to model complex entities and their relationships. While such structured storage can support explicit multi-hop reasoning, it comes at the cost of high construction and maintenance overheads. Furthermore, the complexity of maintaining these intricate topologies often leads to diminishing returns in open-ended dialogue, where flexibility and low response latency are paramount. In contrast, QueryLink challenges the assumption that "structure is everything" in memory systems. Our empirical results demonstrate that a lightweight, flat memory architecture—when enhanced with robust semantic alignment outperforms complex, graph-based models on benchmarks such as LoCoMo. This suggests that, for long-term memory, the alignment mechanism itself is the key factor in improving performance, rather than relying only on memory structure.

3 Method

3.1 Problem Formulation

We formalize long-term interaction as a sequential decision process. Let $\mathcal{D}_t = \{u_1, r_1, \ldots, u_{t-1}, r_{t-1}\}$ denote the dialogue history up to turn t, where u_i represents a user query and r_i represents the agent's response. Due to the context window constraints of LLMs, the complete dialogue history is segmented and stored in an external memory bank $\mathcal{M} = \{m_1, m_2, \ldots, m_N\}$. Given the current query u_t, the goal of the retriever $\mathcal{R}$ is to identify an optimal subset $\mathcal{M}^* \subset \mathcal{M}$ that maximizes the probability of generating the correct response r_t:

$$\mathcal{M}^* = \arg\max_{\mathcal{M}' \subset \mathcal{M}} P(r_t \mid u_t, \mathcal{M}') \tag{1}$$

In standard RAG systems, retrieval typically relies on cosine similarity within a pre-trained embedding space(Gao et al., 2023b). However, due to the mismatch between the sparse, interrogative query space and the dense, declarative memory space, direct similarity measures often fail to accurately capture true relevance. To this end, QueryLink leverages alignment mapping functions Φ_Q and Φ_M to project queries and memories into a shared semantic space $\mathcal{H}$. This semantic alignment in the new aligned space ensures that the similarity measure more reliably reflects true relevance.

3.2 The Architecture of QueryLink

As illustrated in Figure 2, the QueryLink framework operates in three distinct phases: Coherent Memory Chunking, Query-Memory Alignment (QM-Align), and Multi-Grained Retrieval. A key advantage of our framework is its memory-system-agnostic nature. Unlike many existing approaches that depend on rigid topological structures (e.g.,

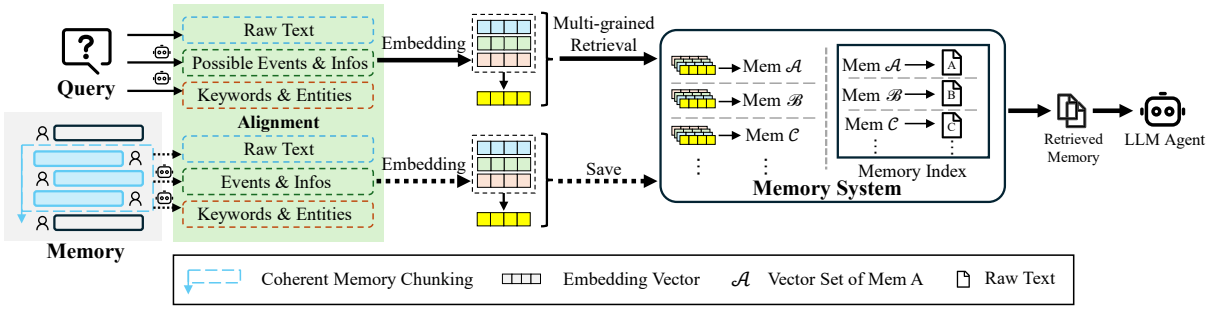

Figure 2: The architecture of QueryLink. It consists of three main components: Coherent Memory Chunking, Query-Memory Alignment, and Multi-Grained Retrieval. Dialogue history is first segmented into semantic units using a sliding window strategy to preserve context. QM-Align projects both memory chunks and queries into symmetric, multi-grained representations (Raw, Events, Keywords), which are then aggregated into a unified Centroid representation. The system retrieves relevant contexts via these multi-grained representations to directly support the LLM agent.

node-edge definitions), our core alignment module (QM-Align) functions independently of the underlying memory storage architecture. This flexibility enables QueryLink to serve as a plug-and-play enhancement across a wide range of retrieval systems—from simple vector stores to complex databases—improving performance without the requirement for significant infrastructure changes.

3.2.1 Coherent Memory Chunking

High-quality semantic alignment fundamentally depends on context-rich information. Traditional RAG systems applied to agents often rely on fixed-size token chunking (e.g., dividing text into 256- or 512-token segments). While computationally efficient, this approach disrupts the semantic continuity of a conversation, leading to the issue of context fragmentation. For example, a standalone utterance like “No, I prefer Python” loses its crucial semantic connection if separated from the preceding question, “Should we use C++?”. This is not a suitable strategy for long-term memory agents.

To address this, we use a dialogue-guided sliding window with a summarization strategy to preserve context coherence. Consecutive multi-turn dialogues are treated as a single semantic unit $C = \{(u_i, r_i), \ldots, (u_{i+W}, r_{i+W})\}$, where W denotes the size of the sliding window. Importantly, when the accumulated context exceeds the window limit, we do not discard older turns. Instead, we employ a dialogue-guided summarization strategy (prompt detailed in Appendix A) to condense key information from the preceding context before appending new turns. This approach ensures that each memory chunk C retains the full "trigger-response" logic while preserving the semantic richness needed for subsequent alignment.

3.2.2 Query-Memory Alignment

Query-Memory Alignment (QM-Align) is the core module of our QueryLink. It employs a symmetric architecture that maps text from both the query side and the memory side into derived semantic subspaces, effectively bridging the semantic gap. For any input text T (whether a user query u or a memory chunk m), we employ an LLM to generate representations at four aligned granularities. This decomposition enables the capture of multiple facets of information, addressing potential limitations that a single vector representation might overlook.

Raw Representation (V_{raw}): This corresponds to the embedding of the original text T, capturing surface-level lexical features and syntax. The raw representation is crucial, as it serves as a baseline grounding, ensuring that the retrieval process stays closely tied to the user’s actual utterance and preventing excessive semantic drift that may arise from over-abstraction in generated representation.

Semantic & Episodic Representation (V_{sem}): To capture deeper intent, we perform asymmetric extraction depending on whether the input is from the query side or the memory side. On the memory side, we prompt the LLM to extract explicit events (e.g., relevant activity) and implicit information (e.g., user preferences inferred from tone). On the query side, we generate hypothetical events and potential intents, effectively transforming an interrogative query into the declarative form of the answer it seeks (e.g., "Try to guess the events or in-

formation related to the ques- tion."). This process can align the sparse, intent-driven query space with the dense, content-rich memory space.

Entity & Keyword Representation (V_{kw}): We extract salient entities, such as proper names, technical terms, and specific locations, which anchor the text in a discrete Lexical Space. While semantic embeddings are powerful, they can sometimes "smooth out" important details. V_{kw} ensures precise matching of proper nouns and technical terms (e.g., error codes or version numbers), which are essential for maintaining accuracy.

Unified Centroid Representation (V_{cen}): After generating the three base representations, we construct a comprehensive unified representation by calculating their geometric centroid as in Eq. (2).

$$\mathbf{V}_{cen} = \text{Normalize}\left(\mathbf{V}_{raw} + \mathbf{V}_{sem} + \mathbf{V}_{kw}\right) \quad (2)$$

We posit that these representations offer complementary perspectives on the underlying semantics. Any single representation consists of the "true semantics" $\mathbf{v}_{truth}$ plus granularity-specific noise ϵ_{noise}. For instance, V_{raw} may contain noise from irrelevant words, while V_{sem} might include hallucinated details. By performing the centroid operation, isotropic noise components from different perspectives tend to cancel each other out (denoising), thereby enhancing the Signal-to-Noise Ratio (SNR) of the core semantic intent.

3.3 Multi-Grained Retrieval Strategy

To maximize Recall in the vast search space of long-term memory, we design a multi-grained retrieval strategy that is "broad in retrieval" based on our QM-Align module. Rather than relying on a single vector similarity, we perform independent retrieval across the four semantic granularities ($\mathcal{V} = \{V_{raw}, V_{sem}, V_{kw}, V_{cen}\}$) and aggregate the results.

$$\mathcal{M}_{retri} = \bigcup_{v \in \mathcal{V}} \text{TopK}(\cos(\mathbf{q}_v, \mathbf{m}_v)) \quad (3)$$

The retrieval operation is described in Eq. (3). We intentionally adopt a union operation rather than an intersection, as different queries may require retrieval from multiple perspectives. The union operation guarantees that relevant memories, even if they don't share significant lexical overlap, are included. This allows the system to capture a broader range of semantic meanings that might be overlooked with a more restrictive intersection approach. Specifically, the V_{kw} representation excels at capturing precise entities; the V_{sem} representation focuses on abstract intents; and the V_{cen} representation provides a solid baseline. By employing the union strategy, QueryLink ensures that memories with high semantic relevance, regardless of their lexical similarity, are retained. This significantly enhances the system's resilience to varying degrees of mismatch.

4 Experiments

4.1 Experimental setup

Datasets. We evaluate QueryLink on two complementary benchmarks to assess its performance across diverse long-context scenarios. Primarily, we use the LoCoMo benchmark (Maharana et al., 2024), which features 10 high-quality, long-horizon conversations averaging 16k tokens each. Its 1,986 questions cover Temporal, Causal, Multi-hop, and Summarization tasks. To further validate the generalizability of QueryLink, we additionally evaluate on LongMemEval (Wu et al., 2024), a comprehensive benchmark comprising 500 meticulously curated questions, each embedded within an extensive and freely scalable user-assistant interaction history. This dataset is specifically designed to evaluate five core long-term memory capabilities in dynamic, task-oriented scenarios: information extraction, multi-session reasoning, temporal reasoning, knowledge updates, and abstention.

Implementation Details. All experiments are conducted using GPT-4o-mini(Achiam et al., 2023) as the backbone LLM, with the embedding model based on all-MiniLM-L6-v2. For memory organization, we implement a coherent chunking strategy with a dynamic window approach, limiting the window size to a maximum of 5 turns. In terms of retrieval hyperparameters, the retrieval candidate size is set to $k = 4$, and the context expansion window is set to $c = 2$, which corresponds to the dynamic window size. To ensure statistical reliability, all reported metrics represent the average of five independent runs.

Baselines. We evaluate QueryLink against leading state-of-the-art methods across three different memory paradigms.

- Full-Context & Compression: LangMem and MemGPT (Packer et al., 2024), which adopt a context-centric strategy by managing long or infinite context through compression and external paging.

Table 1: Comparison with SOTA Methods on the LoCoMo Benchmark. Performance is evaluated based on reasoning accuracy (Judge) and retrieval overlap (F1/BLEU), with the best results highlighted in bold.

Method	Multi-Hop			Temporal			Open Domain			Single Hop			Average		
	J	F1	B1	J	F1	B1	J	F1	B1	J	F1	B1	J	F1	B1
MemoryBank	-	5.00	4.77	-	9.68	6.99	-	5.56	5.94	-	6.61	5.16	-	6.89	5.52
LangMem	-	36.03	27.22	-	38.10	32.23	-	29.79	23.17	-	41.72	35.61	-	39.18	32.59
MemGPT	-	26.65	17.72	-	25.52	19.44	-	9.15	7.44	-	41.04	34.34	-	33.18	26.51
A-MEM	-	27.02	20.09	-	45.85	36.67	-	12.14	12.00	-	44.65	37.06	-	39.65	32.31
Nemori	65.30	36.50	25.60	71.00	56.70	46.60	44.80	20.80	15.10	82.10	54.40	43.20	74.38	49.51	38.93
Mem0	67.13	38.72	27.13	55.51	48.93	40.51	51.15	28.64	21.58	72.93	47.65	38.72	66.88	45.10	35.90
Mem0g	65.71	38.09	26.03	58.13	51.55	40.28	47.19	24.32	18.82	75.71	49.27	40.30	68.44	46.14	36.34
QueryLink (Ours)	**70.21**	**39.13**	**30.67**	**80.69**	**63.90**	**51.03**	**65.63**	**30.38**	**25.06**	**87.87**	**56.96**	**51.19**	**81.75**	**53.48**	**45.77**

- Vector-based RAG: MemoryBank (Zhong et al., 2024) and A-MEM (Wang et al., 2024), which are standard dense retrieval baselines relying on embedding similarity and conventional memory update mechanisms.
- Structured Memory: Mem0 (Chhikara et al., 2025) and Nemori (Nan et al., 2025), which organize memory through knowledge graphs or hypergraphs to facilitate structured retrieval. A comparison with these methods also highlights the contrast between complex structured memory and our well-aligned flat memory approach.

Evaluation Metrics. We report three key metrics: F1 Score, BLEU-1 (B1), and LLM-as-a-Judge (J). The F1 Score measures the lexical overlap between the generated response and the ground truth. B1 evaluates the quality of the generated responses by assessing word overlap with the ground truth, helping to gauge the lexical precision of the output. The J metric (Li et al., 2024) uses GPT-4o to determine whether the generated answer aligns with the ground truth through binary annotation. This metric has a stronger correlation with human judgment in open-ended QA tasks.

4.2 Comparisons with SOTA methods

Comparison with Baselines. Table 1 presents a comprehensive performance comparison on the LoCoMo benchmark. QueryLink sets a new state-of-the-art across all evaluated categories, achieving an average Judge Score of 81.75, significantly outperforming the leading graph-based baselines, Nemori (74.38) and Mem0 (66.88).

Specifically, QueryLink demonstrates a significant advantage in *Temporal* tasks, outperforming Mem0 by over 25% in Judge Score (80.69 vs. 55.51). Graph-based baselines typically represent time as node attributes, which often become disconnected from the narrative flow during retrieval. QueryLink overcomes this limitation through the event representation (V_{sem}), which explicitly extracts and indexes precise timestamps (e.g., "Tuesday at 7 PM") that are tightly associated with their corresponding actions. This approach enables the retriever to identify events not only by keywords but also by their temporal "fingerprint," effectively filtering out irrelevant past or future occurrences that share similar lexical features. In the noise-heavy *Open Domain* category, QueryLink achieves a Judge Score of 65.63, significantly outperforming Nemori (44.80) by 10.83%. Open-ended dialogues often feature informal language, slang, and digressions that challenge structured parsers. QueryLink's unified centroid representation (V_{cen}) plays a key role here: by combining the vector representations of raw text, keywords, and semantics, it effectively reduces granularity-specific noise (denoising), producing a robust signal that remains stable even when user queries are vague or colloquially phrased.

Robustness on LongMemEval. The effectiveness of our alignment mechanism is further corroborated by results on the LongMemEval benchmark (Table 2). QueryLink consistently achieves superior performance across diverse session types and reasoning tasks, scoring significantly higher in Temporal Reasoning (63.16), Multi-Session (55.64) and Assistant (98.21) categories compared to baselines. This consistent gain across different benchmarks suggests that our multi-grained semantic views, derived from domain-agnostic prompts, are robust and transferable to various long-term memory distributions.

Flat vs. Structured Memory. A key insight from our results is the re-examination of memory topology. While prior studies often argue that complex graph structures are essential for multi-hop reasoning, QueryLink offers an efficient alternative. It outperforms Mem0 even in the *Multi-Hop* category, achieving a Judge Score of 70.21 compared to

Table 2: Experimental results (LLM-as-a-Judge) on the LongMemEval benchmark. Bold values indicate the best performance.

Method	Multi-Session	Assistant	Knowledge-Update	User	Temporal	Preference	Avg
Zep	47.40	75.00	**74.40**	92.90	54.10	**53.30**	63.80
Nemori	51.10	83.90	61.50	88.60	61.70	46.70	64.20
QueryLink (Ours)	**55.64**	**98.21**	69.23	**95.71**	**63.16**	50.00	**69.80**

Table 3: Ablation study of Query-Memory Alignment on the LoCoMo benchmark. We report performance using LLM-as-a-Judge (J), F1, and BLEU-1 (B1), with the best results highlighted in bold.

Alignment		Multi-Hop			Temporal			Open Domain			Single Hop			Average		
M-Side	Q-Side	J	F1	B1	J	F1	B1	J	F1	B1	J	F1	B1	J	F1	B1
×	×	45.74	23.97	15.70	52.02	42.15	31.80	48.96	24.03	20.06	65.76	42.17	37.51	58.18	37.70	31.24
×	✓	61.35	33.03	22.74	59.19	48.95	38.15	53.13	21.02	16.80	76.69	50.57	44.64	68.76	45.18	37.54
✓	×	48.23	26.12	16.70	71.34	56.64	43.44	54.17	25.75	22.09	70.63	46.47	41.16	65.65	43.57	35.97
✓	✓	**70.21**	**39.13**	**30.67**	**80.69**	**63.90**	**51.03**	**65.63**	**30.38**	**25.06**	**87.87**	**56.96**	**51.19**	**81.75**	**53.48**	**45.77**

Mem0's 67.13. The limitation of graph-based methods stems from their *schema rigidity* and *construction noise*: converting fluid dialogues into discrete triples often leads to information loss or incorrect links. In contrast, QueryLink's QM-Align mechanism establishes a "soft" alignment in the vector space, allowing the model to implicitly traverse reasoning paths through semantic proximity rather than relying on explicit edges. This reflects that a well-aligned flat memory is sufficient for handling complex reasoning.

4.3 Ablation Study

In this section, we present a comprehensive ablation study to assess the contributions of our components in QueryLink. First, we evaluate the Query-Memory Alignment and Multi-Grained Retrieval Strategy within our multi-grained retrieval, analyzing how different granularities influence performance. Next, we investigate the impact of Coherent Memory Chunking by varying the retrieval count and context window size, analyzing the importance of context continuity and balanced retrieval for effective reasoning.

Effectiveness of query-memory alignment. To validate the contribution of our alignment components, we conduct an ablation study by selectively removing representations from the Memory Side (M-Side) or Query Side (Q-Side). The results in Table 3 show that removing alignment from either side causes a sharp decline in performance. Notably, the full dual-side model achieves an average Judge Score of 81.75, outperforming the baseline without alignment (×/×) by a significant margin of 23.57% points. This confirms that mismatch is a bidirectional issue. Optimizing only one side (either query expansion or memory structuring) creates a "semantic blind spot," whereas symmetric optimization fully aligns the feature spaces.

Effectiveness of multiple granularities. To assess the contribution of multiple granularities, we conduct an ablation study by selectively removing different granularities in the multi-grained retrieval process, with the results presented in Table 4. In this study, each row removes one type of representation—raw text, entity, event, or unified representation—while retaining the others to measure its impact. The results show that removing any single granularity consistently leads to a decrease in performance across all tasks. Our full model, which incorporates all four granularities, achieves the best overall scores, highlighting the importance of multi-granularity representations for robust multi-grained retrieval.

Effectiveness of coherent memory chunking. We investigate the impact of the retrieval count k and context window size c in our coherent memory chunking, as shown in Table 5. The results highlight the critical role of context continuity. Removing context ($c = 0$) results in a significant accuracy drop of 11.36%, as many queries rely on *co-reference resolution* (e.g., resolving terms like "it" or "that meeting"). Expanding the window to $c = 2$ restores these dependencies, providing the generator with self-contained semantic units. Regarding k, we observe a "Curse of Excessive Retrieval." While $k = 8$ slightly benefits Multi-hop reasoning, it degrades performance in Temporal (-5.92%) and Open Domain (-4.17%) tasks. We hypothesize that excessive retrieval introduces *dis-*

Table 4: Ablation Study of Multi-Grained Retrieval Strategy on the LoCoMo Benchmark. We report performance using LLM-as-a-Judge (J), F1, and BLEU-1 (B1), with the best results highlighted in bold.

Strategy	Multi Hop			Temporal			Open Domain			Single Hop			Average		
	J	F1	B1	J	F1	B1	J	F1	B1	J	F1	B1	J	F1	B1
w/o Raw	63.48	33.68	22.14	75.70	61.82	49.56	61.46	27.42	21.82	87.04	56.59	50.94	78.77	51.67	43.56
w/o Entity	67.73	37.87	30.15	75.39	60.64	48.31	61.46	**30.48**	**25.33**	86.56	56.86	51.10	79.22	52.53	45.08
w/o Event	65.60	34.68	25.71	76.64	61.43	49.11	57.29	27.41	22.21	85.02	54.61	49.00	77.99	50.69	43.09
w/o Unified	67.02	37.14	28.35	76.64	62.36	49.96	62.50	28.89	22.98	85.73	56.42	50.90	78.96	52.41	44.83
Ours	**70.21**	**39.13**	**30.67**	**80.69**	**63.90**	**51.03**	**65.63**	30.38	25.06	**87.87**	**56.96**	**51.19**	**81.75**	**53.48**	**45.77**

Table 5: Ablation study of retrieval number k and context window Size c on LoCoMo. The default setting ($k = 4, c = 2$) achieves the best trade-off. The best results in each setting are bolded.

Settings		Multi-Hop			Temporal			Open Domain			Single Hop			Average		
k	c	J	F1	B1	J	F1	B1	Acc	F1	B1	J	F1	B1	J	F1	B1
2	2	68.09	36.28	28.31	77.26	61.75	48.34	57.29	26.45	21.87	85.14	55.93	50.45	78.64	51.71	44.17
4	0	65.25	34.89	26.83	70.72	57.66	45.06	56.25	23.38	19.35	73.60	46.97	42.12	70.39	45.51	38.52
4	1	**70.92**	35.62	26.86	78.19	62.08	49.19	58.33	26.80	23.17	85.26	57.07	51.23	79.48	52.30	44.59
4	2	70.21	**39.13**	**30.67**	**80.69**	**63.90**	**51.03**	**65.63**	30.38	**25.06**	87.87	56.96	51.19	**81.75**	53.48	45.77
8	2	78.37	41.69	34.67	74.77	62.25	49.94	61.46	**31.08**	24.38	**88.70**	**57.17**	**51.37**	82.21	**53.77**	**46.33**

tractor passages, which compete for the LLM's attention, leading to hallucinations or confusion. Based on these findings, we select $k = 4$ and $c = 2$ as the balanced default.

4.4 Efficiency and Latency Analysis

We evaluate the operational efficiency of QueryLink from both online interaction and offline construction perspectives, as detailed in Table 6 and Table 7.

Table 6: Efficiency comparisons against SOTA methods. Latency represents the total end-to-end time per query in milliseconds (ms).

Method	J ↑	Avg. Tokens ↓	Latency (ms) ↓
FullContext	72.30	23,653	5,806
LangMem	51.30	125	22,082
Mem0	66.88	1,027	1,440
Mem0g	68.44	3,616	2,590
RAG-4096	0.30	3,430	2,884
Nemori	74.40	2,745	3,053
QueryLink (Ours)	**81.75**	2,135	2,338

Table 7: Token analysis (in thousands) on LoCoMo during offline memory construction. Data (include J) marked with * is from LightMem (Fang et al., 2025).

Method	J	Sum (in)	Sum (out)	Upd (in)	Upd (out)	Total
LangMem*	57.20	-	-	898.27	111.95	1010.22
A-MEM*	64.16	182.74	49.29	729.89	187.52	1149.43
Mem0	66.88	851.32	20.53	632.12	189.42	1693.39
QueryLink (Ours)	**81.75**	118.12	442.37	-	-	**560.49**

Online and Offline Costs. In terms of online query overhead, QueryLink reduces token usage compared to many agentic frameworks. While QM-Align introduces generative steps for indexing, it is faster than complex graph-based models like Nemori and Mem0g in end-to-end latency. Crucially, our approach yields significant savings during offline memory construction. By bypassing the need to extract complex structural triples or manage high-order relationships, QueryLink consumes only 560.49k tokens for indexing on LoCoMo—substantially lower than the 1693.39k tokens required by Mem0. This highlights that a well-aligned flat memory architecture offers a Pareto-optimal balance between high reasoning accuracy and low maintenance overhead.

Latency and Throughput. In terms of inference speed, *QueryLink* (2,338 ms) is significantly faster than Nemori (3,053 ms) and Mem0g (2,590 ms). While FullContext achieves high precision, its latency (5,806 ms) and extremely high token cost (23,653) make it impractical for real-time applications. Although LangMem is faster in some phases, its performance (51.3 score) falls short for complex memory tasks. *QueryLink* strikes a Pareto-optimal balance, delivering the highest LLM score (81.75) with competitive latency and low cost.

4.5 Transferability Analysis

A core advantage of QueryLink is the modularity of its alignment mechanism. To validate its transferability, we apply our QM-Align module directly to A-MEM (Wang et al., 2024), a standard vector-based baseline. Table 8 provides a detailed performance breakdown across all reasoning cat-

Table 8: Transferability analysis. We integrate our QM-Align module into A-MEM (Wang et al., 2024), resulting in substantial gains in both F1 and BLEU-1 (B1) across all categories.

Method	Metric	Multi	Temp	Open	Single	Avg
A-MEM	F1	27.02	45.85	12.14	44.65	39.65
	B1	20.09	36.67	12.00	37.06	32.31
+ QM-Align	**F1**	**33.93**	**49.99**	**26.35**	**50.55**	**45.88**
	B1	**24.45**	**39.67**	**22.18**	**45.17**	**38.80**

egories. As shown, the integration consistently improves performance. Notably, in the challenging *Open Domain* category, the F1 score more than doubles (from 12.14 to 26.35), demonstrating that our alignment mechanism effectively denoises unstructured contexts where standard bi-encoders struggle. Additionally, while the original A-MEM baseline lacks the *LLM-as-a-Judge* metric for direct comparison, our enhanced model achieves an impressive average Judge Score of 69.87. This score confirms that QM-Align successfully "upgrades" a simple vector database with high-level reasoning capabilities, without modifying the underlying storage infrastructure. Additionally, we verify the scalability of our approach across different backbone models (e.g., Llama-3, Qwen-2.5). Due to space constraints, the detailed multi-model results are provided in Appendix B.

5 Conclusion

This work addresses the challenge of mismatch in long-term memory agents, where interrogative queries often fail to align with declarative historical records. We propose a novel framework, QueryLink, which bridges this semantic gap through multi-grained Query-Memory Alignment. To ensure semantic continuity within memory, we also leverage a dialogue-guided memory chunking method that supports high-quality alignment. By mapping both queries and memories to symmetric semantic subspaces, QueryLink enables robust, multi-faceted retrieval, capturing not only explicit entities but also implicit intents. Extensive evaluations on the LoCoMo and LongMemEval benchmark show that QueryLink outperforms existing state-of-the-art approaches. Our results also challenge the necessity of complex memory topologies, showing that a well-aligned flat architecture can provide a more efficient alternative. Additionally, the adaptability of QueryLink to standard vector-based systems positions it as a plug-and-play component for enhancing long-term reasoning in LLM-based agents.

Limitations

Despite the strong performance of QueryLink, we acknowledge several limitations that merit further investigation.

Inference Overhead. Unlike standard RAG systems that rely solely on efficient embedding lookups, our QM-Align mechanism involves invoking an LLM to generate semantic views (e.g., events, intents) for both queries and memory chunks. Although our experiments demonstrate that QueryLink achieves state-of-the-art efficiency—maintaining significantly lower overall latency and total token costs compared to complex structured memory baselines—the generative nature of our alignment explicitly increases the output tokens during the offline indexing phase (as reflected by the relatively high *Sum (out)* metrics). This specific generation overhead leaves room for optimization. Future iterations could mitigate this by distilling the view generation capability into smaller, specialized encoders, pushing the efficiency closer to pure dense retrieval.

Heuristic Dependencies. Our current framework relies on fixed heuristics for two key components: the retrieval hyperparameters ($k = 4, c = 2$) and the sliding window summarization strategy. While our sensitivity analysis confirms these settings are robust for general benchmarks, they may not be optimal for all dialogue types. For instance, scenarios with extremely interleaved topics or rapid context switching might compromise the integrity of our window-based chunking. Furthermore, different query types might benefit from dynamically adjusted retrieval scopes, which our current static parameter approach does not support.

Acknowledgment

This work was supported by the Natural Science Foundation of China (Grant NO. 62472025).

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. 2025. Mem0: Building production-ready ai agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*.

Felipe De Brigard, Sharda Umanath, and Muireann Irish. 2022. Rethinking the distinction between episodic and semantic memory: Insights from the past, present, and future. *Memory & Cognition*, 50(3):459–463.

Jizhan Fang, Xinle Deng, Haoming Xu, Ziyan Jiang, Yuqi Tang, Ziwen Xu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, and 1 others. 2025. Lightmem: Lightweight and efficient memory-augmented generation. *arXiv preprint arXiv:2510.18866*.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023a. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 535–549.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. 2023b. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2(1).

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. 2025. From rag to memory: Non-parametric continual learning for large language models. *arXiv preprint arXiv:2502.14802*.

Xun Jiang, Feng Li, Han Zhao, Jiahao Qiu, Jiaying Wang, Jun Shao, Shihao Xu, Shu Zhang, Weiling Chen, Xavier Tang, and 1 others. 2024. Long term memory: The foundation of ai self-evolution. *arXiv preprint arXiv:2410.15665*.

Gibbeum Lee, Volker Hartmann, Jongho Park, Dimitris Papailiopoulos, and Kangwook Lee. 2023. Prompted LLMs as chatbot modules for long open-domain conversation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4536–4554, Toronto, Canada. Association for Computational Linguistics.

Haitao Li, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye, and Yiqun Liu. 2024. Llms-as-judges: a comprehensive survey on llm-based evaluation methods. *arXiv preprint arXiv:2412.05579*.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

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.

Jiayan Nan, Wenquan Ma, Wenlong Wu, and Yize Chen. 2025. Nemori: Self-organizing agent memory inspired by cognitive science. *arXiv preprint arXiv:2508.03341*.

Charles Packer, Vivian Fang, Shishir G Patil, Kevin Lin, Sarah Wooders, and Joseph E Gonzalez. 2024. Memgpt: Towards llms as operating systems. In *Thirty-seventh Conference on Neural Information Processing Systems*.

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.

Rana Salama, Jason Cai, Michelle Yuan, Anna Currey, Monica Sunkara, Yi Zhang, and Yassine Benajiba. 2025. Meminsight: Autonomous memory augmentation for llm agents. *arXiv preprint arXiv:2503.21760*.

Haoran Sun and Shaoning Zeng. 2025. Hierarchical memory for high-efficiency long-term reasoning in llm agents. *arXiv preprint arXiv:2507.22925*.

Liang Wang, Nan Yang, and Furu Wei. 2023. Query2doc: Query expansion with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9414–9423.

Weizhi Wang, Li Dong, Hao Cheng, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. 2024. Augmenting language models with long-term memory. *arXiv preprint arXiv:2401.12932*.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. 2024. Longmemeval: Benchmarking chat assistants on long-term interactive memory.

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

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, and 1 others. 2025. The rise and potential of large language model based agents: A survey. *Science China Information Sciences*, 68(2):121101.

Xuan Xiong, Huan Liu, Li Gu, Zhixiang Chi, Yue Qiu, Yuanhao Yu, and Yang Wang. 2026. Etr: Entropy trend reward for efficient chain-of-thought reasoning. *arXiv preprint arXiv:2604.05355*.

Guibin Zhang, Muxin Fu, Guancheng Wan, Miao Yu, Kun Wang, and Shuicheng Yan. 2025a. G-memory: Tracing hierarchical memory for multi-agent systems. *arXiv preprint arXiv:2506.07398*.

Houston H Zhang, Tao Zhang, Baoze Lin, Yuanqi Xue, Yincheng Zhu, Huan Liu, Li Gu, Linfeng Ye, Ziqiang Wang, Xinxin Zuo, and 1 others. 2025b. Widget2code: From visual widgets to ui code via multimodal llms. *arXiv preprint arXiv:2512.19918*.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, 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 19668–19676.

A Prompt Templates

To facilitate reproducibility, we provide the specific prompt templates used in QueryLink. We present them in three categories: Granularity Generation (for indexing), Query Analysis (for retrieval), and Response Generation.

A.1 Context Summarization (Sliding Window)

Table 9 presents the prompt used in our dialogue-guided sliding window strategy (Section 3.2.1) to maintain semantic continuity.

Table 9: Prompt for Context Summarization (Sliding Window Strategy).

Context Summarization Prompt
Summarize the main content of the preceding text. Keep the information you deem important in a concise yet comprehensive manner. Respond with a paragraph. **Preceding Text:** *{context_text}*

A.2 Granularity Generation (Memory Side)

Table 10 details the prompts used to distill raw memory chunks into different semantic granularities.

Table 10: Prompts for generating Memory-Side Granularities (V_{kw} and V_{sem}).

1. Keyword Extraction (V_{kw})
Summarize the keywords and entities in the following content: *{input_content}* Respond with a list of keywords and entities only, separated by commas.
2. Event Extraction (V_{sem} **- Explicit**)
Summarize the events in the following content: *{input_content}* Respond with a list of events only, each event start with a hyphen.
3. Implicit Information Extraction (V_{sem} **- Implicit**)
Summarize the important information in the following content whatever you can reasonably extract: *{input_content}* Respond with a list of important information only, each info start with a hyphen.

A.3 Query Analysis (Query Side)

Table 11 shows the prompts used to map user queries into the memory space.

Table 11: Prompts for generating Query-Side Granularities (Intent & Hypothetical Events).

1. Query Keywords (V_{kw})
Summarize the keywords and entities in the following question for retrieval: *{user_query}* Answer with a list of keywords and entities only, separated by commas.
2. Hypothetical Event Generation (V_{sem})
Try to guess the events or information related to the question: *{user_query}* Answer with a list of at most three possible relevant events or information only, each answer start with a hyphen.

A.4 Response Generation

Once the relevant memory chunks are retrieved, we construct a structured context and generate the final answer.

Context Integration. We organize retrieved chunks by their session ID to preserve local dialogue structure. The context template for each session is formatted as follows:

- Retrieved Information {ID}:
- date of dialog: {date}
- speaker {role}: {content}
...

Answer Synthesis. Table 12 presents the prompt for generating the comprehensive answer, and Table 13 shows the prompt for refining the answer into a brief format for evaluation.

B Scalability across Backbone Models

In the main experiments, we utilized GPT-4o-mini. To evaluate the robustness of QueryLink, we extended our evaluation to three diverse backbone models: **GPT-4o** (SOTA proprietary), **Llama-3.2-3B**, and **Qwen-2.5-3B** (lightweight open-source).

Table 14 presents the comprehensive results. The comparison reveals that while baseline methods (MemoryBank, MemGPT, A-MEM) often suffer performance collapse on smaller models (e.g., Llama-3B), ours maintains remarkable stability and

Table 12: Prompt for Comprehensive Response Generation.

System Prompt
Based on the following filtered retrieved information, question date (if exists) and question, provide a comprehensive answer. If you can't reason about the answer, please just answer "not mentioned" or "no answer". The answer should be an absolute value, not a relative value. For example, the answer should not be "last year", but a specific year. **Retrieved Information:** *{filtered_info}* **Question Date:** *{question_date}* **Question:** *{question}* **Your Response:**

Table 13: Prompt for Answer Refinement (Detailed to Brief).

System Prompt
Provide a brief and concise answer to the question based on the following detailed answer. Complete sentences are not required; just answer the question directly. **Example:** - Question: "When was the company founded?" Detailed Answer: "The company was founded in 1998 by a group of entrepreneurs." Brief Answer: "1998" ... [More Examples] ... **Question:** *{question}* **Detailed Answer:** *{detailed_response}* **Brief Answer:**

significantly outperforms all baselines across all categories and backbones.

Table 14: Experimental results on the LoCoMo dataset across four reasoning categories (Multi Hop, Temporal, Open Domain, Single Hop). Results are reported in **F1** and **BLEU-1** scores. The best performance in each group is marked in **bold**, and our proposed method demonstrates superior generalizability across diverse foundation models.

Model		Method	Category								Average	
			Multi Hop		Temporal		Open Domain		Single Hop		Average	
			F1	BLEU	F1	BLEU	F1	BLEU	F1	BLEU	F1	BLEU
GPT-4o		MemoryBank	6.49	4.69	2.47	2.43	6.43	5.30	8.28	7.10	6.63	5.57
		MemGPT	30.36	22.83	17.29	13.18	12.24	11.87	60.16	53.35	42.78	36.80
		A-MEM	32.86	23.76	39.41	31.23	17.10	15.84	48.43	42.97	41.75	35.31
		Ours	**41.40**	**33.78**	**64.07**	**53.06**	**29.89**	**25.07**	**55.97**	**49.92**	**53.37**	**46.07**
Llama 3.2	3B	MemoryBank	6.19	4.47	3.49	3.13	4.07	4.57	7.61	6.03	6.27	5.05
		MemGPT	5.32	3.99	2.68	2.72	5.64	5.54	4.32	3.51	4.24	3.56
		A-MEM	17.44	11.74	26.38	19.50	12.53	11.83	28.14	23.87	24.84	19.99
		Ours	**23.42**	**15.99**	**40.56**	**31.25**	**21.94**	**17.80**	**42.18**	**33.29**	**37.15**	**28.73**
Qwen 2.5	3B	MemoryBank	3.60	3.39	1.72	1.97	6.63	6.58	4.11	3.32	3.68	3.25
		MemGPT	5.07	4.31	2.94	2.95	7.04	7.10	7.26	5.52	5.94	4.86
		A-MEM	12.57	9.01	27.59	25.07	7.12	7.28	17.23	13.12	17.91	14.49
		Ours	**22.64**	**14.79**	**23.70**	**18.74**	**17.10**	**15.69**	**39.13**	**33.69**	**31.52**	**25.99**