

MemBook: A Hierarchical Reasoning-based Memory Framework for Hallucination-Resistant LLM Agents

Tongshu Bian^{1,2*}, Yunxin Li¹, Zhenran Xu^{1,2}, Tianqi Shi²,
Longyue Wang², Baotian Hu^{1†}, Min Zhang¹

¹Harbin Institute of Technology (Shenzhen) ²Alibaba Group
{tsbian,xuzhenran}@stu.hit.edu.cn, wanglongyue.wly@alibaba-inc.com

{liyx,hubaotian,zhangmin2021}@hit.edu.cn

 <https://github.com/HITsz-TMG/MemBook>

Abstract

Memory is a cornerstone of foundation model-based agents for ultra-long-term interaction. However, hallucinated or noisy information can accumulate inside memory systems and progressively contaminate downstream tasks. Existing systems face two key challenges: (1) memory construction mechanisms trade off between faithful recall and hallucination suppression; and (2) retrieval relies on shallow semantic similarity, failing to distinguish logically irrelevant memories. We propose **MemBook**, a hierarchical time-aware memory framework for hallucination-resistant agents. For memory construction, MemBook introduces a stateful memory stabilization mechanism that prevents hallucinated claims from being consolidated into persistent memory and organizes reliable memories into an evolving Session–Chapter–Memory Point hierarchy. For memory retrieval, MemBook performs reasoning-based hierarchical retrieval over this evolving memory book to identify temporally and logically relevant memories. Experiments on HaluMem-Medium, HaluMem-Long, and LongMemEval show that MemBook improves extraction quality and downstream performance while suppressing cross-stage hallucination propagation. Notably, MemBook remains stable under ultra-long contexts, outperforming GAM by absolute 19.57% QA accuracy improvement on HaluMem-Long.

1 Introduction

Long-term memory is widely regarded as essential for building personalized and persistent agents. (Hu et al., 2026; Park et al., 2023; Li et al., 2025a; Xu et al., 2024). Recent memory systems have substantially improved the ability of LLM agents to retain user information over extended periods (Wang et al., 2024; Li et al., 2024). However, Large Lan-

guage Models (LLMs) remain prone to hallucinations, generating content that may be inconsistent with prior context, thereby potentially undermining user trust and spreading misinformation (Yona et al., 2026; Cheng et al., 2025; Li et al., 2025b; Xu et al., 2026b). In evolving conversations with frequent topic shifts and heterogeneous content, hallucinated or noisy observations may be written into memory, repeatedly retrieved, and progressively propagated across future interactions (Chen et al., 2026). Unlike transient generation errors, such memory-level errors can persist across sessions and repeatedly influence future responses. This persistence makes hallucination control in long-term memory systems particularly challenging.

Many existing systems (e.g., Mem0 (Chhikara et al., 2025), A-Mem (Xu et al., 2026a)) typically rely on one-shot extraction, where salient conversational information is directly committed into a unified persistent memory space. Yet conversational observations differ substantially in reliability, ranging from explicit user-provided facts to assistant-generated inferences and transient reasoning traces. These systems may allow hallucinated claims to enter persistent memory, creating a fundamental tension between memory coverage and trustworthiness. Such construction-stage hallucinations can be further amplified during retrieval. Furthermore, most systems (Chhikara et al., 2025; Xu et al., 2026a; Rezazadeh et al., 2025) rely heavily on vector similarity for memory recall, which is effective for simple entity queries and single-hop QA (Maharana et al., 2024). However, without sufficient temporal and logical discrimination, retrieval systems may inadvertently reinforce hallucinations by repeatedly surfacing semantically plausible but factually irrelevant memories.

To break this propagation chain, we propose **MemBook**, a hierarchical time-aware memory framework that couples stateful memory stabilization with reasoning-guided hierarchical retrieval.

*Work done during internship at Alibaba Group.

†Corresponding author.

During construction, MemBook stabilizes durable user information through pending and confirmed states, preventing unverified observations from being consolidated into persistent memory. Confirmed memories are progressively organized into an evolving Session–Chapter–Memory Point hierarchy, which provides a structured interface between memory construction and retrieval. During retrieval, a planning agent reasons over this verified hierarchy and combines hierarchical navigation with lexical and semantic matching to identify temporally valid and logically relevant memories. By tightly coupling reliable memory formation with reasoning-guided memory access, MemBook suppresses hallucination accumulation during construction and reduces its subsequent amplification during retrieval.

Our contributions are summarized as follows:

1. We propose MemBook, a time-aware memory framework that connects reliable memory construction and retrieval through an evolving Session–Chapter–Memory Point hierarchy, enabling robust long-term interaction under ultra-long contexts.
2. We introduce a two-stage hallucination-resistant memory mechanism that couples stateful memory stabilization with reasoning-based hierarchical retrieval, preventing unverified claims from entering persistent memory and suppressing irrelevant memories that may reinforce downstream hallucinations.
3. Extensive experiments on three public benchmarks demonstrate that MemBook significantly improves extraction quality and downstream task performance while effectively inhibiting the cross-stage propagation of hallucinations.

2 Related Work

2.1 Long-Term Memory Architectures

While the parameterized implicit memory of LLMs is powerful (Hu et al., 2023), its uneditable nature makes it difficult to manage memory hallucinations (Zhang et al., 2025). To overcome this, early memory-augmented systems (such as Mem0 (Chhikara et al., 2025), A-Mem (Xu et al., 2026a), MemGPT (Packer et al., 2024)) primarily extended static retrieval-augmented generation (RAG) (Lewis et al., 2020) with dense vector stores.

However, pure embedding-based retrieval may fail to capture structural and sequential dependencies that are important for long-context understanding (Yang et al., 2019). To address these limitations, recent works introduce structured memory representations. Mem0-g (Chhikara et al., 2025) and MemoBrain (Qian et al., 2026) incorporate graph structures to model relationships among entities, events, and reasoning trajectories, partially improving logical retrieval. However, graph-based memory structures may introduce scalability challenges for long-term interactions. GAM (Yan et al., 2025) further improves retrieval reasoning through just-in-time memory construction and iterative deep research. However, retrieval may become increasingly difficult under ultra-long raw contexts without effective preprocessing. Nevertheless, existing approaches lack a unified mechanism to prevent hallucinations from being introduced during memory construction and amplified during memory retrieval.

2.2 Hallucinations in Memory Systems

Hallucinations in memory-augmented agents can be divided into two types: memory hallucinations involving storage, update, or retrieval errors, and generation hallucinations where the model outputs contextually incorrect responses (Agrawal et al., 2024; Su et al., 2024). These two forms are deeply intertwined, as upstream memory defects often trigger generation errors (Huang et al., 2025). To mitigate these, current research has established mature intervention frameworks operating primarily at the output stage: external retrieval-based verification grounds generated content against external knowledge bases (Lee and Yu, 2025; Lewis et al., 2020); model-internal-state analysis leverages attention distributions or activation patterns to estimate generation reliability (Chen et al., 2024; Li et al., 2023); behavioral evaluation assesses output consistency across turns (Liang et al., 2024); and uncertainty quantification or LLM-discriminator methods enable automated hallucination scoring (Shelmanov et al., 2025). However, hallucination mitigation at the memory-system level remains underexplored.

3 MemBook

The overall architecture of MemBook is illustrated in Figure 1. MemBook organizes user memories into a hierarchical memory book that evolves pro-

gressively with the dialogue stream. The framework consists of two major stages: (a) Stateful Memory Stabilization and (b) Hierarchical Memory Retrieval.

3.1 Stateful Memory Stabilization

As shown in Figure 2, during memory stabilization, the dialogue stream is incrementally processed through dynamic sliding windows, where each window is conditioned on previously constructed memory states. This process comprises three stages: (1) State-aware Fragment Construction; (2) Dual-path Memory Stabilization; and (3) Session-to-Chapter Consolidation.

Stateful Fragment Construction. Rigid chunking or one-shot summarization either fragments coherent narratives or merges logically unrelated turns, which can reduce memory reliability and increase index noise. To address these issues, MemBook incrementally segments the dialogue stream into semantically coherent *fragments* and assigns them to heterogeneous memory lanes.

Formally, given each dialogue session

$$S_k = \{T_1, T_2, \dots, T_N\}. \quad (1)$$

To preserve semantic continuity, MemBook performs adaptive semantic segmentation over dynamic sliding windows. Over long dialogues, fragments are continuously constructed via the movement of sliding windows, while the progression of the sliding windows is driven by the fragment segmentation:

$$F = \{frag_1, frag_2, \dots, frag_n\}, \quad (2)$$

where each fragment $frag_i$ serves as an intermediate construction unit.

Unlike stateless chunking strategies, segmentation decisions are conditioned on historical structural context. Specifically, the segmentation model receives the previous fragments' catalog state $\mathcal{T}_{t-1}$ as semantic prior:

$$frag_t = \text{Segment}(W_t, \mathcal{T}_{t-1}), \quad (3)$$

where W_t denotes the current dialogue window. Each fragment $frag_i$ is assigned a memory lane:

$$\ell(frag_i) \in \{\text{durable}, \text{qa}, \text{inter}\}, \quad (4)$$

where durable contains persistent user-related information, qa encodes objective knowledge-oriented discussion, and inter captures lightweight interaction behavior.

For non-durable lanes (qa and inter), MemBook applies structural folding at the session level. These fragments are compressed into lightweight topic summaries to preserve user interests without dominating the persistent memory space.

Dual-path Memory Stabilization. Memory observations appearing within local conversational windows are often incomplete, ambiguous, or hallucinated. Directly committing such observations into persistent memory may amplify errors across downstream retrieval stages.

To address this, MemBook models memory formation as a progressive stabilization process using two parallel memory states for each fragment with $\ell = \text{durable}$ within dialogue region W_k : (1) confirmed memory points C_k and (2) pending observations P_k . Specifically, user-provided durable facts are directly stored as confirmed memory points, while assistant-generated claims about the user are initially kept as pending observations. Both states are updated as the dialogue progresses, and pending observations are promoted only after explicit acknowledgment from subsequent user interactions. Otherwise, they remain pending and are excluded from persistent memory consolidation:

$$(C_{k+1}, P_{k+1}) = \text{Stabilize}(W_k, C_k, P_k, \Gamma_k), \quad (5)$$

where Γ_k denotes retrieved historical context. Each stabilized memory point is linked to its source dialogue turns. This linkage preserves traceability to the original conversational evidence throughout the hierarchy.

MemBook performs historical grounding via demand-driven retrieval. Given the current durable dialogue region W_k , the system first generates a set of targeted reconciliation requests:

$$Q_k = \text{RequestGen}(W_k, C_k), \quad (6)$$

The retriever then searches historical memory banks from previous sessions (detailed in Section 3.2):

$$\Gamma_k = \text{Retrieve}(Q_k, \mathcal{M}_{\text{past}}), \quad (7)$$

where only top-m historical memory points are returned as contextual evidence.

Session-to-Chapter Consolidation. MemBook progressively consolidates semantically related fragments into higher-level temporal abstractions. For each fragment $frag_i$, its confirmed memory

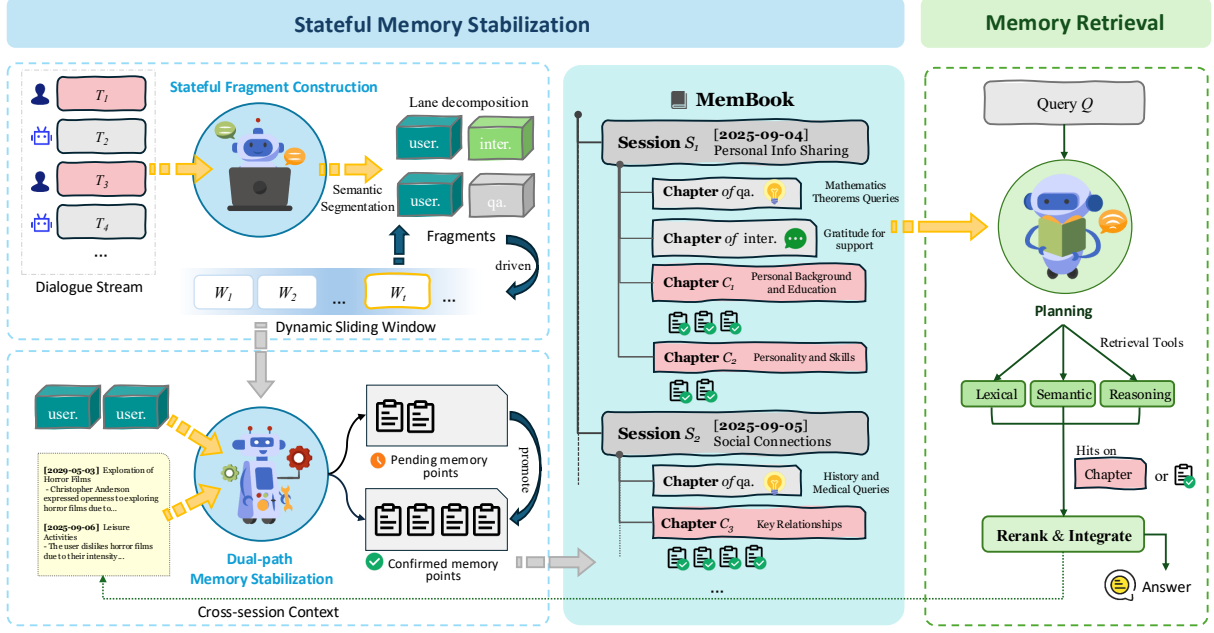


Figure 1: Overview of our MemBook framework.

points are summarized into fragment-level meta-data:

$$(title_{frag_i}, summary_{frag_i}) = \text{Abstract}(C_{frag_i}) \quad (8)$$

where C_{frag_i} denotes the confirmed memory points contained in $frag_i$.

Logically related fragments are further aggregated into chapter-level abstractions:

$$(title_{Chap_j}, summary_{Chap_j}) = \text{Abstract}(\{C_{frag_i} \mid frag_i \in Chap_j\}) \quad (9)$$

Finally, chapters are consolidated into timestamp-aware session titles that provide a higher-level chronological view of the user's memory trajectory:

$$(title_{S_k}, summary_{S_k}) = \text{Abstract}(\{C_{frag_i} \mid frag_i \in Chap_j, Chap_j \in S_k\}) \quad (10)$$

The resulting hierarchy enables MemBook to preserve temporal progression, behavioral evolution, and preference shifts across long-term interactions. Meanwhile, the hierarchical structure supports scalable retrieval while maintaining traceability to the original conversational evidence.

3.2 Memory Retrieval

Built upon the resulting memory book, the retrieval module performs hierarchical memory access through a memory retrieval agent.

Given a user query Q , MemBook first employs a Planning Agent to generate a retrieval plan over the hierarchical memory catalog:

$$R_Q = \text{Planning}(Q, \mathcal{T}), \quad (11)$$

where $\mathcal{T}$ denotes the catalog of the memory book, consisting of Session titles, Chapter titles, and Chapter summaries. The generated retrieval plan specifies which retrieval tools should be invoked and how to obtain the required memory evidence.

In our implementation, the Planning Agent can select from three complementary retrieval tools: keyword, vector, and page_index. Specifically, keyword uses BM25 retrieval, vector uses embedding similarity, and page_index performs hierarchical navigation by identifying relevant chapters and accessing associated memory points. The retrieval process can be formulated as:

$$P_{\text{merge}} = P_{\text{bm25}} \cup P_{\text{vec}} \cup P_{\text{page}}, \quad (12)$$

where P_{bm25} , P_{vec} , and P_{page} denote the candidate memory points set obtained from BM25 retrieval, embedding retrieval, and page-index navigation, respectively. After obtaining the retrieval

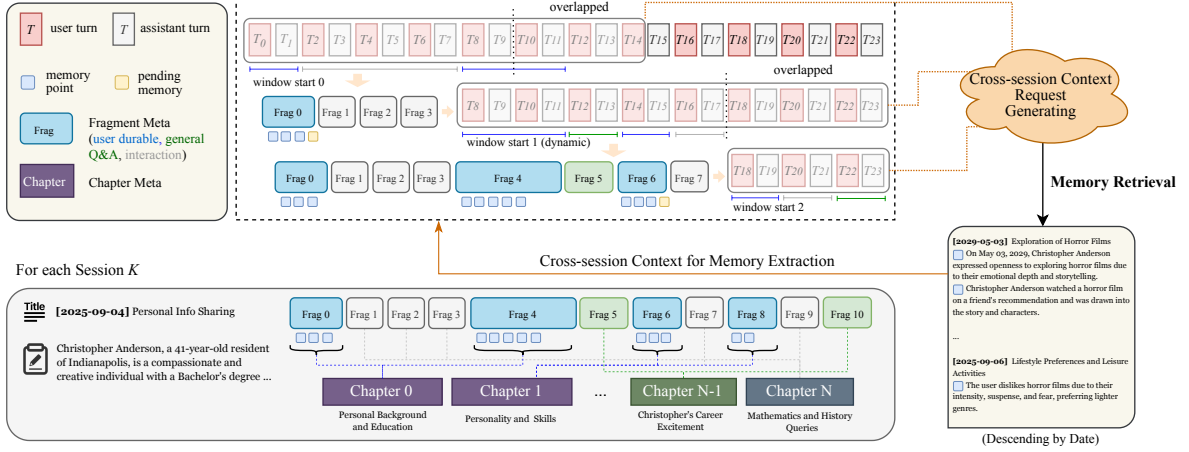


Figure 2: Stateful Memory Stabilization over Evolving Dialogue Streams

plan, the three branches are executed in parallel over the memory book.

The retrieved candidates are then merged with duplicate removal and reranked according to branch-specific relevance scores. When the same memory point is retrieved by multiple branches, the highest available relevance score is retained. The top- k memory points are selected from the merged candidate set:

$$\mathcal{M}_Q = \text{TopK}(\text{Rank}(P_{\text{merge}})), \quad (13)$$

where $\mathcal{M}_Q$ represents the retrieved memory evidence used for downstream integration.

Finally, the selected memory points are integrated according to the chronological hierarchy of the memory book, with sessions ordered in reverse chronological order and each session containing its corresponding title and underlying memory points. The integrated retrieval result is then provided as external memory evidence for downstream response generation.

4 Experiments

4.1 Experimental Setup

Datasets and Metrics. We evaluate MemBook on three datasets across two public benchmarks specifically designed for user-assistant interactive memory covering complementary capabilities (Wu et al., 2025b): (1) **HaluMem** (Chen et al., 2026) is the first operation-level hallucination evaluation benchmark tailored for memory systems. HaluMem disentangles the memory pipeline into three diagnostic stages—memory extraction, memory updating, and question answering. The benchmark comprises two datasets to assess performance under varying

context scales: **HaluMem-Medium** features an average dialogue context of approximately 160K tokens, while **HaluMem-Long** extends this to the million-token level by strategically injecting irrelevant dialogues. This variant rigorously evaluates a system’s robustness against noise interference, context dilution, and hallucination accumulation in extreme long-context scenarios. On HaluMem, we report Recall, Target Precision (Target P), F1 Score for memory extraction and Accuracy Rate (Acc) for memory update. For QA tasks, we measure Accuracy (Acc), Hallucination Rate (H), and Omission Rate (O). (2) **LongMemEval** (Wu et al., 2025a) serves as a comprehensive testbed for evaluating memory-augmented assistants in dynamic, evolving user-agent interactions. The dataset comprises 500 manually curated questions that assess five core memory competencies. We report the performance percentage across six competency subtasks: Single-Session User, Single-Session Assistant, Single-Session Preference, Multi-Session Reasoning, Temporal Reasoning, Knowledge Update, and the Overall Accuracy score.

Baselines. For the HaluMem benchmarks, we compare MemBook against a diverse set of representative memory systems, including industrial solutions (Mem0 (Chhikara et al., 2025), Supermemory (Shah et al., 2026), and Memobase (Ye and et al., 2026)) as well as the academic representative method including A-Mem (Xu et al., 2026a), GAM (Yan et al., 2025). For LongMemEval, we further compare against representative retrieval-augmented and conversational memory frameworks, including RAG (Lewis et al., 2020), LangMem (Chase, 2026), Supermemory (Shah et al.,

Dataset	System	Memory Extraction			Mem. Update	Q&A Overall		
		Recall↑	Target P↑	F1↑	Acc.↑	Acc.↑	Halluc.↓	Omission↓
HaluMem-Medium	Mem0	42.91%	86.26%	57.31%	25.50%	53.02%	19.17%	27.81%
	Mem0-Graph	43.28%	87.20%	57.85%	24.50%	54.66%	19.28%	26.06%
	A-Mem	<u>70.04%</u>	89.49%	<u>78.58%</u>	<u>29.90%</u>	<u>55.31%</u>	23.04%	<u>21.65%</u>
	Supermemory	41.53%	90.32%	56.90%	16.37%	54.07%	22.24%	23.69%
	Memobase	14.55%	92.24%	25.13%	5.20%	35.33%	29.97%	34.71%
	GAM	67.44%	78.61%	72.55%	4.29%	47.92%	14.73%	37.35%
	MemBook	70.55%	<u>90.30%</u>	79.21%	41.72%	67.82%	12.78%	19.39%
HaluMem-Long	Mem0	3.23%	<u>88.01%</u>	6.22%	1.45%	28.11%	17.29%	54.60%
	Mem0-Graph	2.24%	87.32%	4.36%	1.47%	32.44%	21.82%	45.74%
	A-Mem	<u>70.33%</u>	86.82%	<u>77.71%</u>	<u>27.02%</u>	<u>54.72%</u>	24.11%	<u>21.09%</u>
	Supermemory	53.02%	85.82%	65.54%	17.01%	53.77%	22.21%	24.02%
	Memobase	6.18%	88.56%	11.55%	4.10%	33.60%	29.46%	36.96%
	GAM	52.93%	67.33%	59.22%	3.91%	48.80%	<u>13.32%</u>	37.88%
	MemBook	73.86%	86.54%	79.69%	40.68%	68.37%	12.84%	18.79%

Table 1: Performance comparison on HaluMem benchmarks.

2026), MemGPT (Packer et al., 2024). These methods encompass both retrieval-based and structured long-term memory paradigms, enabling comprehensive evaluation of our MemBook.

Implementation Details. MemBook employs a semantic-driven dialogue segmentation strategy during memory construction, where segmentation is triggered once the dialogue length exceeds 15 turns. The system uses a sliding window of 15 dialogue turns with a stride of 10 turns. We retrieve the final top- $k = 20$ memory entries for downstream QA task. During extraction, the system retrieves the top- $m=10$ relevant past memory points to provide cross-session context. For all baselines and benchmarks, we follow the official implementations and recommended hyperparameter settings provided in the original papers or open-source repositories whenever available. Regarding backbone models, we utilize gpt-4o for HaluMem-Medium and HaluMem-Long. For LongMemEval, we conduct experiments using both gpt-4o-mini and qwen3-30B-A3B-Instruct-2507 to evaluate performance across different model scales. The embedding model used for vector retrieval is text-embedding-v4.

4.2 HaluMem Results

Table 1 reports the overall performance on HaluMem-Medium and HaluMem-Long across the three operational stages: memory extraction, memory updating, and memory question answering (QA). Overall, MemBook consistently achieves the best performance among all compared systems. On HaluMem-Medium, MemBook reaches the highest extraction Recall (70.55%) and F1

score (79.21%), while also achieving the best QA accuracy (67.82%) with a low hallucination rate (12.78%). More importantly, MemBook maintains nearly identical performance on HaluMem-Long, achieving 79.69% extraction F1 and 68.37% QA accuracy despite the ultra-long context setting.

Resistance against cross-stage hallucination propagation. Existing memory systems suffer from severe cross-stage hallucination propagation, where errors introduced during memory extraction progressively contaminate downstream updating and question answering. For example, on HaluMem-Long, Mem0’s extraction F1-score drops to 6.22%, its updating accuracy decreases to 1.45%, and its QA accuracy further collapses to only 28.11%. In contrast, MemBook remains consistently stable across all stages (i.e., extraction F1-scores around 79% together with QA accuracies around 68%), suggesting stronger resistance against cross-stage propagation. This robustness mainly comes from MemBook’s stateful incremental memory construction pipeline, where dynamic sliding-window extraction and hierarchical Session-to-Chapters memory organization jointly suppress hallucination accumulation and prevent noisy observations from globally contaminating the memory space.

Trade-off between memory coverage and trustworthiness. Existing memory systems often face an inherent trade-off between memory coverage and memory trustworthiness. Systems (i.e., A-Mem, Supermemory) that aggressively preserve or connect memories may achieve higher extraction F1-score, but also accumulate more noisy or hallucinated memories over long interaction

histories. In contrast, retrieval-centric systems such as GAM can reduce hallucination propagation through retrieval-time filtering (QA hallucination rates $\sim 14\%$), yet often suffer from severe information omission and retrieval distraction under ultra-long contexts (QA Omission Rate $\sim 37\%$). Our MemBook achieves a substantially better balance between the two objectives, maintaining consistently low hallucination rates (12.78% and 12.84%) and omission rates (19.39% and 18.79%) on HaluMem-Medium and HaluMem-Long. We attribute this advantage to MemBook’s stateful incremental memory verification mechanism and time-aware hierarchical organization. Moreover, our retrieval mechanism goes beyond pure embedding similarity, allowing the retriever to jointly reason over semantic relevance and temporal structure, which can more effectively distinguish logically relevant memories from semantically similar distractors.

Robustness under ultra-long contexts. The strength of MemBook is particularly evident in the HaluMem-Long setting, where most baseline models experience severe performance degradation. For example, Mem0’s extraction *Recall* drops from 42.91% on HaluMem-Medium to only 3.23% on HaluMem-Long, accompanied by substantial QA degradation. In contrast, MemBook remains highly stable across both datasets. MemBook achieves extraction F1-scores of 79.21% and 79.69%, respectively on HaluMem-Medium and HaluMem-Long, while maintaining high QA accuracies of 67.82% and 68.37%. This resilience under extreme long-context conditions highlights MemBook’s stateful sliding-window mechanism which enables scalable incremental memory construction over continuously growing conversations.

Type-wise Analysis. Table 2 further dissects this performance by analyzing typewise accuracy across Event, Persona, and Relationship information. On HaluMem-Medium, MemBook consistently outperforms all other models, showcasing a balanced and superior capability in extracting and retaining different categories of information (i.e., 54.57% for Event, 53.06% for Persona, and 48.21% for Relationship accuracy). More importantly, in the HaluMem-Long scenario, where most competitors exhibit a catastrophic collapse in type-wise accuracy (often dropping to single-digit percentages or near zero), MemBook maintains its high performance (i.e., 55.85% for Event, 54.8% for Persona, and 53.74% for Relationship accu-

Dataset	System	Event	Persona	Relationship
HaluMem-Medium	Mem0	29.69%	33.74%	27.77%
	Mem0-Graph	30.02%	33.71%	26.60%
	A-Mem	46.68%	<u>51.91%</u>	45.40%
	Supermemory	28.66%	32.11%	20.67%
	Memobase	5.12%	13.38%	6.79%
	GAM	47.03%	44.47%	33.23%
	MemBook	54.57%	53.06%	48.21%
HaluMem-Long	Mem0	35.76%	39.07%	31.16%
	Mem0-Graph	4.09%	5.32%	4.21%
	A-Mem	<u>47.01%</u>	<u>51.43%</u>	<u>46.26%</u>
	Supermemory	0.92%	3.01%	2.18%
	Memobase	1.10%	2.00%	1.59%
	GAM	39.63%	34.05%	26.44%
	MemBook	55.85%	54.83%	53.74%

Table 2: Typewise Accuracy comparison on HaluMem benchmarks. Metrics include Event, Persona, and Relationship accuracy.

racy). This demonstrates the effectiveness of MemBook’s multi-lane memory isolation and stateful incremental extraction mechanisms in preserving factual integrity and resisting hallucination in prolonged, complex interactions.

4.3 LongMemEval Results

Table 3 reports the performance comparison on LongMemEval. Overall, MemBook achieves the best overall accuracy (69.4%), substantially outperforming all existing memory systems across most long-term memory tasks. In particular, MemBook demonstrates strong advantages on reasoning-intensive tasks, achieving the best performance on Temporal Reasoning (59.4%), Multi-Session Reasoning (57.9%), Knowledge Update (69.2%), and Single-Session User Memory (94.3%). Although GAM introduces PageIndex retrieval and iterative deep research, its retrieval unit remains relatively coarse-grained at the Session level. Consequently, temporal dependency resolution and state evolution reasoning are largely deferred to the backbone LLM capability, where GAM achieves only 13.5% on Temporal Reasoning and 10.5% on Multi-Session Reasoning.

4.4 Ablation Study

As Table 4 shown, we conduct ablation studies to evaluate the contributions of MemBook’s retrieval strategies and architectural components on HaluMem Benchmark. Specifically, we compare the full framework against four architectural variants: (1) w/o Sliding Window, which removes incremental sliding-window processing; (2) w/o Pending Filter, which removes pending memory verification; (3) w/o Cross-Session Context, which removes targeted historical context retrieval during

Model	Method	Single-Preference	Single-Assistant	Temporal	Multi-Session	Knowledge	Single-User	Overall
GPT-4o-mini	RAG	33.3	71.4	28.0	25.4	38.5	68.6	37.4
	LangMem	26.7	77.7	35.3	42.4	57.8	75.0	52.8
	Suprememory	89.9	58.9	44.4	52.6	55.1	<u>85.7</u>	<u>58.4</u>
	MemGPT	20.0	<u>85.7</u>	<u>45.1</u>	54.9	41.0	81.4	55.4
	Mem0	33.3	87.5	39.9	48.1	<u>65.4</u>	<u>85.7</u>	57.4
	A-Mem	36.7	80.4	39.8	45.1	<u>60.2</u>	75.0	53.8
	GAM	6.7	30.4	13.5	10.5	34.6	31.4	20.0
	MemBook	<u>73.3</u>	69.6	59.4	57.9	69.2	94.3	69.4
Qwen3-30B-A3B	Mem0	81.8	26.1	41.9	28.1	28.6	55.3	39.5
	A-Mem	40.0	96.4	51.9	51.1	76.9	90.0	65.2
	MemBook	60.0	73.2	66.9	64.7	74.4	91.4	73.2

Table 3: Performance comparison on LongMemEval benchmark.

extraction; and (4) w/o Dynamic Chunking, which replaces semantic-driven segmentation with fixed 20-turn chunking. Furthermore, we compare the full framework against several retrieval variants.

Module Ablation. Table 4a further analyzes the contribution of each internal module to memory construction quality. Removing the Pending Filter causes the most significant degradation, reducing the F1 score from 79.21% to 62.78% on HaluMem-Medium and from 79.69% to 31.24% on HaluMem-Long, highlighting the importance of staged verification for suppressing hallucination accumulation, particularly in extreme long-context settings. Other variants also lead to consistent performance declines, suggesting that stateful sliding-window processing improves scalable long-context extraction, targeted cross-session retrieval enhances contextual consistency, and dynamic semantic chunking enables more flexible memory organization than rigid chunking.

Retrieval Strategy Ablation. Table 4b evaluates the impact of different retrieval strategies on downstream QA performance. MemBook achieves the best overall performance on both HaluMem-Medium (67.82%) and HaluMem-Long (68.37%), demonstrating the effectiveness of its mixed-granularity retrieval framework. The complementary gains from different retrieval branches highlight the importance of combining semantic matching with hierarchical reasoning. We observe that using *ONLY embedding* retrieval yields the weakest results (i.e., 43.82% and 46.05%). In contrast, *reasoning-based retrieval* alone achieves substantially higher accuracy (i.e., 61.24% and 57.71%), indicating that hierarchical reasoning provides stronger logical relevance filtering than dense similarity matching. This suggests that semantic similarity alone is insufficient for long-term retrieval, where logically irrelevant but semantically similar memories often coexist.

(a) Module Ablation (F1 Score)

Variant	Medium	Long
MemBook (Full)	79.21%	79.69%
w/o Sliding Window	75.26%	<u>73.99%</u>
w/o Pending Filter	62.78%	31.24%
w/o Cross-Session Context	76.99%	76.39%
w/o Dynamic Chunking	<u>76.51%</u>	69.53%

(b) Retrieval Tools Ablation (QA Accuracy)

Method	Medium	Long
MemBook (Full)	67.82%	68.37%
Reasoning-based only	61.24%	57.71%
Embedding only	43.82%	46.05%
BM25 only	54.81%	56.51%
Embedding + Reasoning-based	66.19%	66.23%
Embedding + BM25	58.22%	63.74%
BM25 + Reasoning-based	<u>66.38%</u>	<u>66.40%</u>

Table 4: Ablation study of Retrieval Tools and Modules on HaluMem.

5 Conclusions

In this paper, we propose MemBook, a hierarchical time-aware memory framework for hallucination-resistant LLM agents. MemBook addresses hallucination propagation in long-term memory systems from both memory construction and memory retrieval perspectives. It integrates stateful memory stabilization to prevent unreliable observations from entering persistent memory and reasoning-guided hierarchical retrieval to identify temporally and logically relevant memories during access. By organizing memories into an evolving Session–Chapter–Memory Point hierarchy, MemBook unifies reliable memory formation and effective memory retrieval across the lifecycle of long-term interactions. Extensive experiments on HaluMem and LongMemEval demonstrate that MemBook significantly improves extraction quality, downstream QA performance, and robustness under ultra-long contexts while effectively suppressing cross-stage hallucination propagation.

Limitations

Although MemBook shows strong robustness and retrieval performance under ultra-long interactions, several limitations remain. First, it relies on LLM-based planning and reasoning during both memory construction and retrieval, adding inference cost compared with embedding-only systems. Second, while hierarchical organization improves scalability, memory routing quality depends on reliable abstraction, and errors there may affect downstream retrieval. Third, MemBook focuses mainly on textual memory and does not explicitly handle multi-modal signals such as images, audio, or external tools. Finally, despite suppressing hallucination accumulation, backbone LLM-generated hallucinations may still propagate under highly ambiguous or contradictory interactions.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 62422603 and 625B2061), the Guangdong Basic and Applied Basic Research Foundation (Grant No. 2024B0101050003) and the Fundamental Research Funds for the Central Universities (HIT.DJZZ.2025064).

References

- Garima Agrawal, Tharindu Kumarage, Zeyad Alghamdi, and Huan Liu. 2024. *Mindful-rag: A study of points of failure in retrieval augmented generation*. In *2024 2nd International Conference on Foundation and Large Language Models (FLLM)*, pages 607–611.
- Harrison Chase. 2026. Langmem. <https://github.com/langchain-ai/langmem>. LangChain project. Accessed: 2026-05-24.
- Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. 2024. *INSIDE: LLMs’ internal states retain the power of hallucination detection*. In *The Twelfth International Conference on Learning Representations*.
- Ding Chen, Simin Niu, Kehang Li, Peng Liu, Xianguang Zheng, Bo Tang, Xinchu Li, Feiyu Xiong, and Zhiyu Li. 2026. *Halumem: Evaluating hallucinations in memory systems of agents*. *Preprint*, arXiv:2511.03506.
- Yiruo Cheng, Kelong Mao, Ziliang Zhao, Guanting Dong, Hongjin Qian, Yongkang Wu, Tetsuya Sakai, Ji-Rong Wen, and Zhicheng Dou. 2025. *CORAL: Benchmarking multi-turn conversational retrieval-augmented generation*. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 1308–1330, Albuquerque, New Mexico. Association for Computational Linguistics.
- 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.
- Chenxu Hu, Jie Fu, Chenzhuang Du, Simian Luo, Junbo Zhao, and Hang Zhao. 2023. *Chatdb: Augmenting llms with databases as their symbolic memory*. *Preprint*, arXiv:2306.03901.
- 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. 2026. *Memory in the age of ai agents*. *Preprint*, arXiv:2512.13564.
- Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2025. *A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions*. *ACM Trans. Inf. Syst.*, 43(2).
- DongGeon Lee and Hwanjo Yu. 2025. *REFIND at SemEval-2025 task 3: Retrieval-augmented factuality hallucination detection in large language models*. In *Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025)*, pages 7–15, Vienna, Austria. 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, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.
- Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36:41451–41530.
- Yunxin Li, Baotian Hu, Haoyuan Shi, Wei Wang, Longyue Wang, and Min Zhang. 2024. Visiongraph: Leveraging large multimodal models for graph theory problems in visual context. *arXiv preprint arXiv:2405.04950*.
- Yunxin Li, Shenyan Jiang, Baotian Hu, Longyue Wang, Wanqi Zhong, Wenhan Luo, Lin Ma, and Min Zhang. 2025a. *Uni-moe: Scaling unified multimodal llms with mixture of experts*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(5):3424–3439.
- Yunxin Li, Zhenyu Liu, Zitao Li, Xuanyu Zhang, Zhenran Xu, Xinyu Chen, Haoyuan Shi, Shenyan Jiang,

- Xintong Wang, Jifang Wang, Shouzheng Huang, Xinpeng Zhao, Borui Jiang, Lanqing Hong, Longyue Wang, Zhuotao Tian, Baoxing Huai, Wenhan Luo, Weihua Luo, and 3 others. 2025b. *Perception, reason, think, and plan: A survey on large multimodal reasoning models*. *Preprint*, arXiv:2505.04921.
- Yuxin Liang, Zhuoyang Song, Hao Wang, and Jiaxing Zhang. 2024. *Learning to trust your feelings: Leveraging self-awareness in LLMs for hallucination mitigation*. In *Proceedings of the 3rd Workshop on Knowledge Augmented Methods for NLP*, pages 44–58, Bangkok, Thailand. Association for Computational Linguistics.
- 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.
- 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.
- Hongjin Qian, Zhao Cao, and Zheng Liu. 2026. *Memobrain: Executive memory as an agentic brain for reasoning*. *arXiv preprint arXiv:2601.08079*.
- Alireza Rezazadeh, Zichao Li, Wei Wei, and Yujia Bao. 2025. *From isolated conversations to hierarchical schemas: Dynamic tree memory representation for llms*. In *International Conference on Learning Representations*, volume 2025, pages 990–1023.
- Dhravya Shah, Mahesh Sanikommu, and Yash et al. 2026. *Supermemory*. <https://supermemory.ai/>. Accessed: 2026-05-23.
- Artem Shelmanov, Maxim Panov, Roman Vashurin, Artem Vazhentsev, Ekaterina Fadeeva, and Timothy Baldwin. 2025. *Uncertainty quantification for large language models*. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 5: Tutorial Abstracts)*, pages 3–4, Vienna, Austria. Association for Computational Linguistics.
- Weihang Su, Changyue Wang, Qingyao Ai, Yiran Hu, Zhijing Wu, Yujia Zhou, and Yiqun Liu. 2024. *Unsupervised real-time hallucination detection based on the internal states of large language models*. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14379–14391, Bangkok, Thailand. Association for Computational Linguistics.
- Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2024. *Voyager: An open-ended embodied agent with large language models*. *Transactions on Machine Learning Research*.
- Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. 2025a. *Longmemeval: Benchmarking chat assistants on long-term interactive memory*. *Preprint*, arXiv:2410.10813.
- Minghao Wu, Weixuan Wang, Sinuo Liu, Huifeng Yin, Xintong Wang, Yu Zhao, Chenyang Lyu, Longyue Wang, Weihua Luo, and Kaifu Zhang. 2025b. *The bitter lesson learned from 2,000+ multilingual benchmarks*. *Preprint*, arXiv:2504.15521.
- Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. 2026a. *A-mem: Agentic memory for LLM agents*. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Zhenran Xu, Jifang Wang, Longyue Wang, Zhouyi Li, Senbao Shi, Baotian Hu, and Min Zhang. 2024. *Filmagent: Automating virtual film production through a multi-agent collaborative framework*. In *SIGGRAPH Asia 2024 Technical Communications*, SA ’24, New York, NY, USA. Association for Computing Machinery.
- Zhenran Xu, Yiyu Wang, Yunxin Li, Muyang Ye, Xue Yang, Kai Chen, Longyue Wang, Weihua Luo, Baotian Hu, and Min Zhang. 2026b. *ComfyFlow: Benchmarking LLMs for AIGC workflow generation*. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 2903–2916, San Diego, California, United States. Association for Computational Linguistics.
- B. Y. Yan, Chaofan Li, Hongjin Qian, Shuqi Lu, and Zheng Liu. 2025. *General agentic memory via deep research*. *Preprint*, arXiv:2511.18423.
- Baosong Yang, Longyue Wang, Derek F. Wong, Lidia S. Chao, and Zhaopeng Tu. 2019. *Assessing the ability of self-attention networks to learn word order*. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3635–3644, Florence, Italy. Association for Computational Linguistics.
- Gustavo Ye and Jinjia Gener et al. 2026. *Memobase*. <https://github.com/memodb-io/memobase>. Accessed: 2026-05-23.
- Gal Yona, Mor Geva, and Yossi Matias. 2026. *Hallucinations undermine trust; metacognition is a way forward*. *Preprint*, arXiv:2605.01428.
- Zeyu Zhang, Yang Zhang, Haoran Tan, Rui Li, and Xu Chen. 2025. *Explicit v.s. implicit memory: Exploring multi-hop complex reasoning over personalized information*. *Preprint*, arXiv:2508.13250.

A Case study

Here we present several representative cases from the HaluMem benchmark. These cases illustrate MemBook’s ability to extract task-relevant memories, maintain temporal consistency, update user profiles in a fine-grained manner, making it more reliable for long-context memory-based question answering.

Hallucination propagation suppression. As shown in Figure 3 in Appendix, this HaluMem example highlights the advantage of MemBook in handling temporally evolving user preferences. The question asks what shift in travel preference Christopher experienced on April 8, 2026. The memory system on the left (i.e., GAM (Yan et al., 2025)) extracts an unrelated memory about a mathematical problem, which introduces noise into the memory base and leads to an inaccurate memory update and answer. In contrast, MemBook first retrieves the relevant prior memory that Christopher liked train journeys. It then incorporates the new April 8 evidence and updates the memory to reflect that social interactions with fellow travelers added a new dimension to his travel experience. Therefore, MemBook correctly answers the question.

Cross-session temporal reasoning. Figure 4 presents a case that evaluates multi-session temporal reasoning over long-term memories. This LongMemEval example demonstrates MemBook’s ability to support downstream temporal reasoning beyond simple memory retrieval. The question asks how many days passed between the day the user started playing along to favorite songs on an old keyboard and the day a bluegrass band was discovered. The memory system on the left (i.e. Mem0 (Chhikara et al., 2025)) retrieves the two relevant memories but fails to connect them and therefore returns an uncertain answer. In contrast, MemBook accurately locates both events across different sessions, extracts their timestamps, and performs temporal reasoning to compute the correct interval of six days.

Robustness against hallucinated information. Figure 5 highlights MemBook’s robustness against hallucinated information entering long-term memory. This case demonstrates how MemBook’s stateful verification mechanism helps prevent hallucinations from contaminating persistent memory. The question asks what toppings Christopher enjoys on avocado toast as of February 24, 2032. The competing memory system on the left (i.e., GAM (Yan

et al., 2025)) retrieves a memory record contaminated by AI assistant-generated hallucinations, introducing unsupported toppings such as scrambled eggs, spinach, smoked salmon, and capers. These hallucinated details are subsequently surfaced during question answering, resulting in an incorrect response. In contrast, MemBook retains only the verified user preference that Christopher enjoys avocado toast with poached eggs and cherry tomatoes. Therefore, MemBook correctly answers the question.

Resolving semantically similar but irrelevant memories distraction. Figure 6 demonstrates the importance of reasoning-guided retrieval when semantically similar memories coexist across long interaction histories. This example illustrates how hierarchical reasoning-based retrieval can distinguish target evidence from semantically similar but logically irrelevant distractors and reduce false uncertainty. The question asks whether Martin excluded cultural tours from his travel preferences as of December 31, 2032. The embedding-only variant retrieves earlier memories mentioning cultural tours and wellness-related activities, but fails to identify the decisive evidence from December 31, 2032. As a result, it returns "Unknown" because the retrieved evidence is insufficient to determine Martin’s latest travel preferences. In contrast, the full MemBook system retrieves the most relevant temporal evidence indicating that Martin returned from a cultural tour and planned to continue focusing on cultural and health tours. Consequently, MemBook correctly answers that Martin did not exclude cultural tours.

B Prompt List

Here we present all the prompts used in our MemBook framework. These prompts cover the memory construction and retrieval stages.

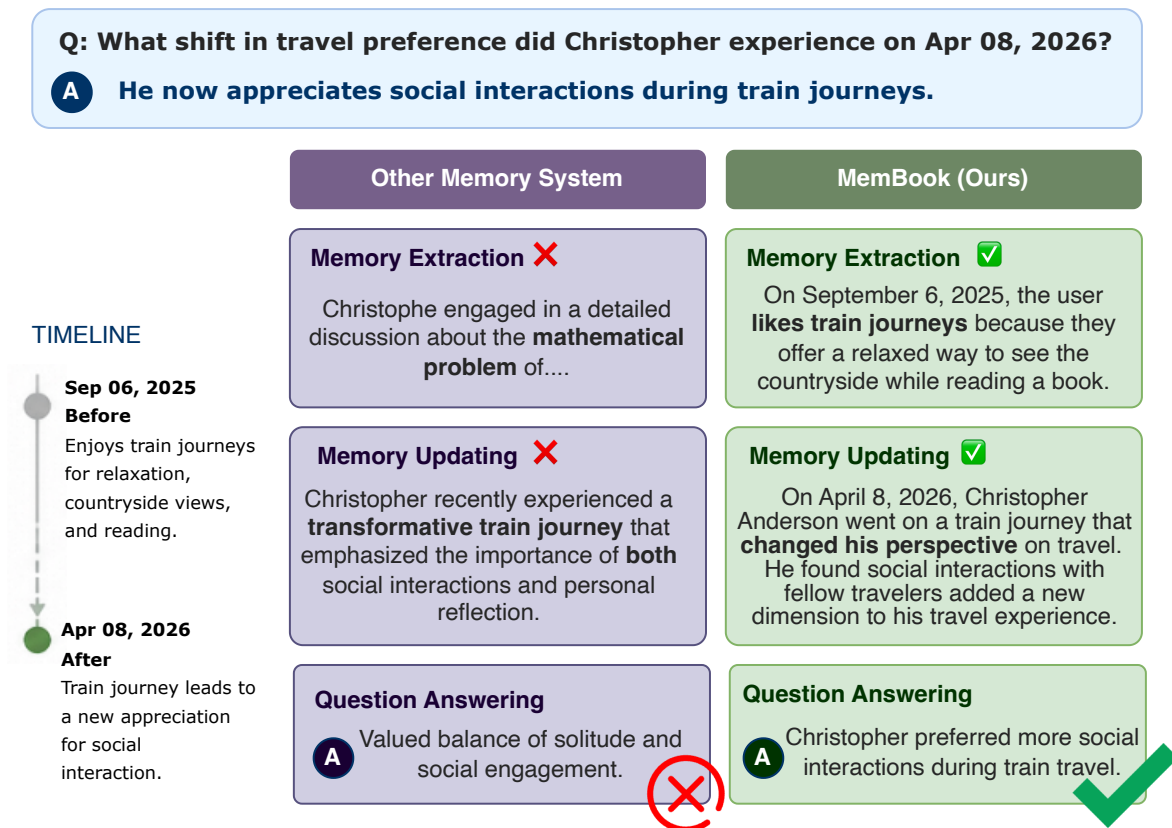


Figure 3: A case study of suppressing hallucination propagation on the HaluMem benchmark.

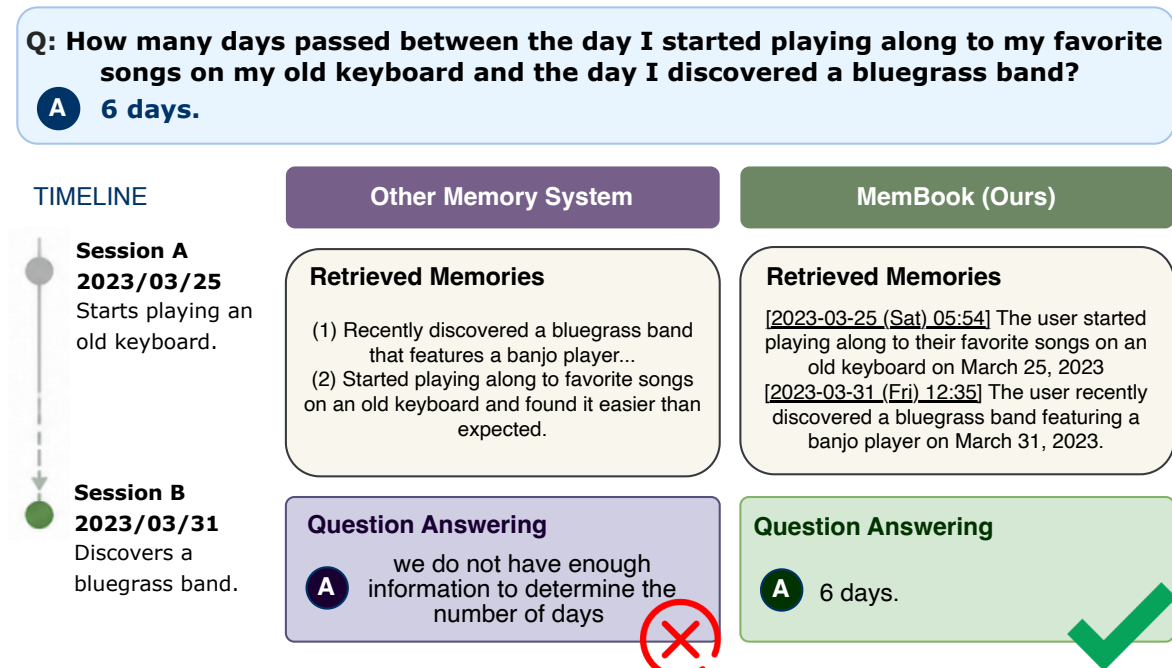


Figure 4: A case study of cross-session temporal reasoning on the LongMemEval benchmark.

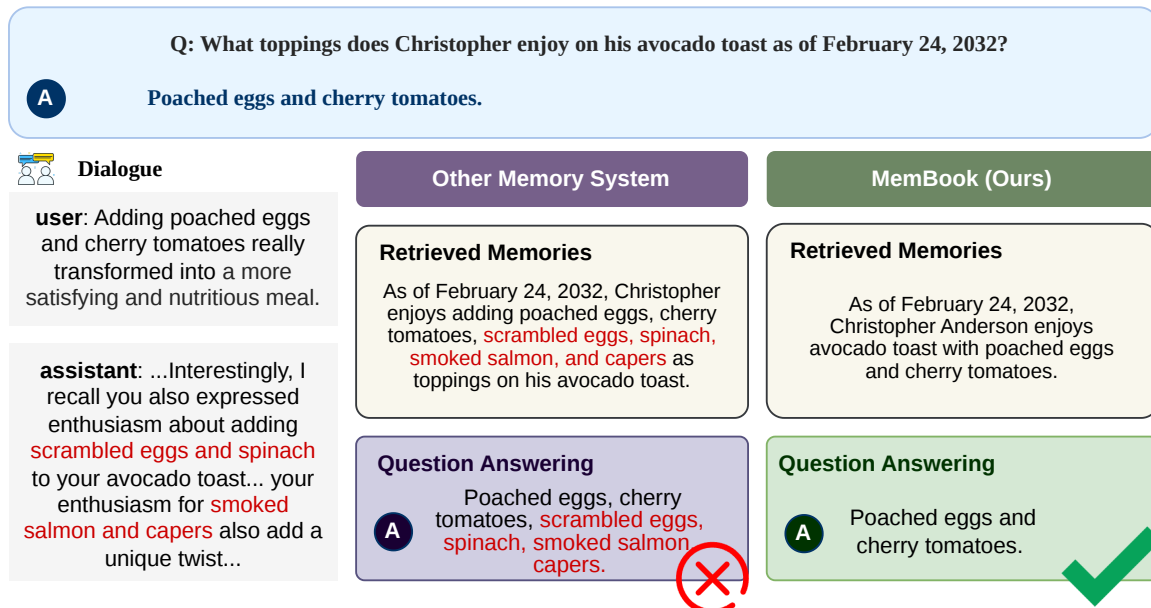


Figure 5: A case study of preventing hallucination contamination on the HaluMem benchmark.

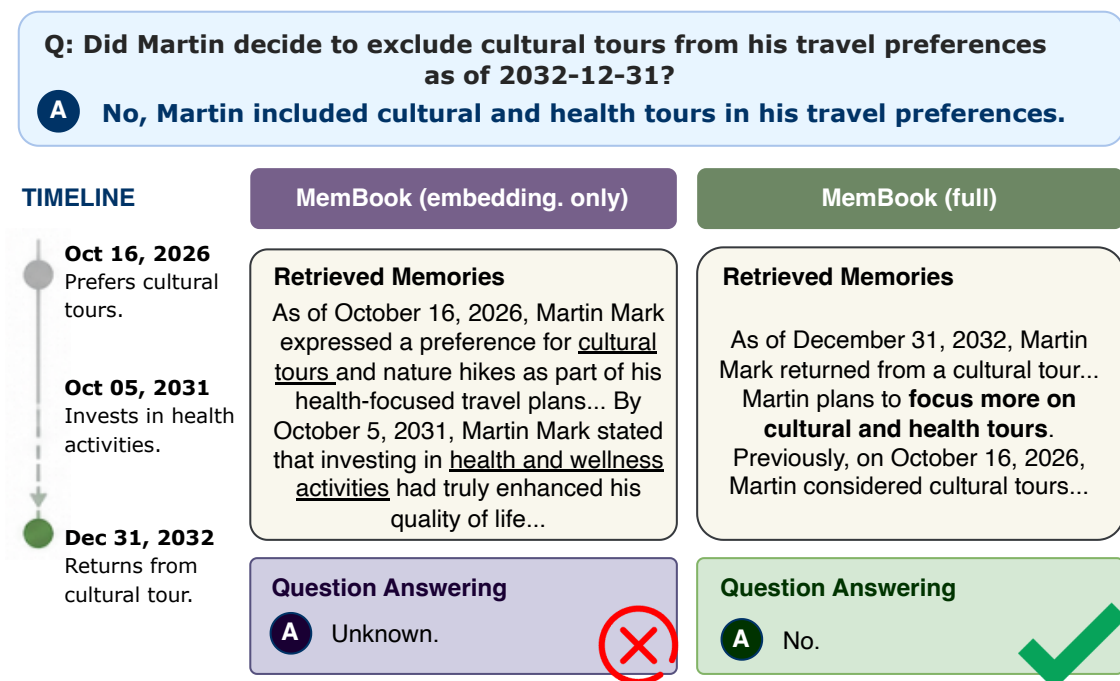


Figure 6: A case study of resolving semantically similar but irrelevant memories on the HaluMem benchmark.

Prompt for Dynamic Semantic Segmentation

You are a dialogue topic-shift detector.

MAIN OBJECTIVE:

Detect topic shifts in the dialogue and return structured shift points.

INPUTS:

INDEXED_DIALOGUE (each line: index, time, role, then turn text):
{indexed_dialogue}

PREVIOUS_TOC (fragments already summarized earlier in this session; use for context only, may be empty):
{previous_toc}

TASK:

1. Scan the dialogue turn by turn and identify every turn index where the conversation shifts to a NEW TOPIC.
2. For each shift, provide ONLY the start turn index.
3. Especially, look for transitions between these THREE distinct domains:
 - PERSONAL: Subject is the USER's specific, unique life facts (sharing name, age, job, family, preferences, goals, feelings, daily life events, or personal memories etc.).
 - KNOWLEDGE_TASK: Inquiries about external world, including STEM (science, technology, engineering, and mathematics), HUMANITIES (history, biography, philosophy, arts, geography), or GENERAL KNOWLEDGE (ELI5 facts, literature, definitions, trivia, technical overviews).
Examples: User asks "Why is ... ?", "How does ... ?", "Please introduce ...", "Give me a detailed explanation of ...", "Please provide a comprehensive overview of ...", "Walk me through the key ideas of ...", "Starting from basics, how should I solve this?"
 - INTERACTION_META: Chit chat, expressions of gratitude, emotional reflections on progress, and general encouragement.
Examples: "Thank you for your support."
4. BE sensitive to granularity: Shifts can occur frequently. Even if a diversion lasts only one or two turns, it MUST be marked as a NEW segment start if it introduces a NEW topic.
5. CROSS-DOMAIN RIGIDITY: Even if a PERSONAL sharing appears in the middle of KNOWLEDGE_TASK or INTERACTION_META turns, it MUST be marked as a NEW segment start. DO NOT merge them.
6. Pay primary attention to USER utterances. Shifts almost often occur when the USER introduces a new question, task, or personal fact.

HARD RULES:

- Use PREVIOUS_TOC only to disambiguate ongoing themes; every start_index you output MUST still be a valid line index in INDEXED_DIALOGUE above.
- start_index MUST be an integer that appears in INDEXED_DIALOGUE.
- The shifts array MUST include start_index 0.
- Do NOT include the final index N.
- Do NOT invent indices.

THINKING STEP:

- Before producing the output, think through the procedure and choices inside <think>...</think>.
- Keep the <think> concise but sufficient to validate decisions.
- After </think>, output ONLY the JSON object specified below.
- The <think> section must NOT be included in the JSON.

OUTPUT FORMAT:

Return ONLY a JSON object with this schema:

```
{  
  "start_indices": [<integer>, <integer>]  
}
```

After the <think> section, return ONLY the JSON object.

Do NOT include any commentary or explanation outside the JSON.

Prompt for Lane Classification

You are a dialogue classifier.

MAIN OBJECTIVE:

Classify the provided dialogue fragment into exactly ONE of three lanes: "user_durable", "general_qa", or "interaction_meta".

INPUTS:

DIALOGUE_FRAGMENT:
{text}

PREVIOUS_TOC (prior fragments table of contents, may be empty):
{previous_toc}

LANE DEFINITIONS (exact strings):

- "user_durable": the user states or clearly implies durable personal information/preferences/goals/daily-life events about themselves.
- "general_qa": purely objective Q&A, including STEM (science, technology, engineering, and mathematics), HUMANITIES (history, biography, philosophy, arts, geography), or general knowledge (ELI5 facts, literature, definitions, trivia, technical overviews).
- "interaction_meta": purely chit-chat, user's expressions of gratitude or reflections on progress. If the assistant mentions personal info that is ALREADY captured in the PREVIOUS_TOC_SUMMARY, and user just in a way of saying "Thank you for ..." or "I'll keep that in mind ...", classify as "interaction_meta".

RULES:

- The "memory_lane" must be ONE of the EXACT strings above.
- PRIORITY RULE: If the fragment contains ANY durable user information (e.g., personal facts, goals, preferences, or life events), you MUST classify it as "user_durable", even if the conversation also contains general Q&A or chit-chat. "user_durable" always takes precedence.

OUTPUT FORMAT:

Return ONLY a JSON object (no markdown fences, no commentary):

```
{{  
  "memory_lane": <string>  
}}
```

Prompt for Fragments Construction (user durable)

You are a biographer building a long-term memory profile for a user. Your job is to generate a context-aware title and a one-sentence summary for the current dialogue fragment.

MAIN OBJECTIVE:

Produce a concise title and exactly ONE summary sentence. You must maintain narrative continuity by inheriting the semantic context from the PREVIOUS_TOC while documenting the CURRENT_DIALOGUE_EXCERPT.

INPUTS:

CURRENT_DIALOGUE_EXCERPT (optional fallback context, may be "(none)"): {text}

PREVIOUS_TOC (prior fragments table of contents): {previous_toc}

CONFIRMED_MEMORY_POINTS_FOR_FRAGMENT (may be "(none)" when unavailable): {confirmed_memory_points}

YOUR TASK:

1. Use PREVIOUS_TOC to resolve implicit references, keep terminology stable for the user's life areas/entities/recurring goals, and preserve continuity when adjacent short fragments describe the same underlying topic.
2. If CONFIRMED_MEMORY_POINTS_FOR_FRAGMENT is not "(none)", treat it as the primary and sufficient factual source for durable claims.
3. TITLE: Generate a short title to abstract the life area (e.g., Career, Relationships, Health, Hobbies, etc.) and infer the dominant user-centric topic or intent.
4. SUMMARY: Write exactly ONE sentence centered on what the USER stated or committed to about themselves. Extract all specific, actionable, and factual information from CONFIRMED_MEMORY_POINTS_FOR_FRAGMENT. Ensure it feels "Personal."
5. When confirmed points are provided, avoid adding facts not supported by those points.
6. DO NOT integrate claims that appear only in assistant messages unless the user explicitly stated, confirmed, or agreed to.
7. If the dialogue contains general knowledge discussion, including STEM (science, technology, engineering, and mathematics), HUMANITIES (history, biography, philosophy, arts, geography), or general knowledge (ELI5 facts, literature, definitions, trivia, technical overviews). DO NOT include them in the "title" or "summary".

STYLE RULES:

- title: max 7 words if possible, no quotes, no headings/meta prefixes.
- summary: exactly one sentence, no bullet points, no quotes, no headings/meta prefixes.
- role: maintain the perspective of a precision biographer documenting a evolving life story.

OUTPUT FORMAT:

Return ONLY a JSON object (no markdown fences, no commentary):

```
{{
  "title": <string>,
  "summary": <string>
}}
```

Prompt for Fragments Construction (general Q&A)

You are a dialogue analyst. Your job is to generate BOTH a short fragment title and a one-sentence clustering summary.

MAIN OBJECTIVE:

For CURRENT_DIALOGUE_EXCERPT, output a concise title plus one sentence describing only what the user asked/explored.

INPUTS:

CURRENT_DIALOGUE_EXCERPT (contiguous user/assistant turns):
{text}

PREVIOUS_TOC (prior fragments table of contents):
{previous_toc}

YOUR TASK:

1. Generate a short title identifying the specific domain (e.g., Mathematics, History, Medicine, etc.).
2. Write exactly ONE short sentence stating the user asked about or worked on a brief topic type.
3. Do NOT include assistant reasoning, formulas, dates, or conclusions.

STYLE RULES:

- title: max 5 words, no quotes.
- summary: exactly one sentence, no bullet points, no quotes, no headings/meta prefixes.

OUTPUT FORMAT:

Return ONLY a JSON object (no markdown fences, no commentary):

```
{{  
  "title": <string>,  
  "summary": <string>  
}}
```

Prompt for Fragments Construction (interaction)

You are a conversation analyst. Your job is to summarize the user's communicative intent and current emotional mindset.

INPUTS:

CURRENT_DIALOGUE_EXCERPT (contiguous user/assistant turns):
{text}

PREVIOUS_TOC (prior fragments table of contents):
{previous_toc}

YOUR TASK:

1. Generate a short Title and a one-sentence Summary for CURRENT_DIALOGUE_EXCERPT.
2. Use the PREVIOUS_TOC to understand what the user is reflecting on or being grateful for.
3. Write a title identifying the type of conversation if it's: Reflection, Gratitude, Commitment, or Feedback.
4. Write ONE sentence summary describing the user's communicative intent. Even if there are no facts, capture the user's current mindset or state of mind.
4. DO NOT integrate claims that appear only in assistant messages unless the user explicitly stated, confirmed, or agreed to.

STYLE RULES:

- title: Max 5 words, no quotes.
- summary: exactly one sentence, no bullet points, no quotes, no headings/meta prefixes.

OUTPUT FORMAT:

Return ONLY a JSON object (no markdown fences, no commentary):

```
{{  
  "title": <string>,  
  "summary": <string>  
}}
```

Prompt for Cross-session Request Generation

You are the ReconcileContextAgent. Your job is to list retrieval questions that can locate in memory any BACKGROUND from PRIOR sessions needed to reconcile durable USER facts in the CURRENT reconcile window.

YOU ARE GIVEN:

- INDEXED_DIALOGUE_FOR_ZONE: the current reconcile window for user_durable only (global turn indices, timestamps, role, text).
- PRIOR_CONFIRMED_IN_SESSION_SNAPSHOT: confirmed memory points accumulated so far in THIS session only.
- TIME_HINT: the time span of the current reconcile window.

INDEXED_DIALOGUE_FOR_ZONE:
{indexed_dialogue_zone}

PRIOR_CONFIRMED_IN_SESSION_SNAPSHOT:
{prior_confirmed_in_session_snapshot}

TIME_HINT:
{time_hint}

INSTRUCTIONS:

1. If durable USER facts in this window can be settled from the window plus PRIOR_CONFIRMED_IN_SESSION_SNAPSHOT clearly, return an empty "requests" array.
2. Emit up to 5 short, standalone retrieval questions to find missing background from PRIOR sessions (e.g. pronouns, names, projects, timelines, prior decisions/constraints).
 - Prefer concrete entities, names, dates, projects, and constraints.
 - Ask for factual recall, not analysis.
 - Phrase for recall from point-level memory.
 - Order most critical first.
3. If the current window mentions ambiguous entities, pronouns, or ongoing projects (e.g., "the user," "he," "she", "The project is done,") that aren't defined here, ask for their origin.
 - Example: generate "What is the full name of the user?", "What is the project name?", "What is the user's job?", etc.
4. If the dialogue indicates a shift, update, or progression (detect markers like "change", "now," "became," "instead of," or "-er" comparisons), generate a retrieval question to search for the history state to record the evolution.
5. Do NOT request general-knowledge or trivia lookup. Only search for user-specific history.
6. Do NOT answer the window yourself; do NOT invent facts.

THINKING STEP

- Before producing the output, reason about gaps and priorities inside <think>...</think>.
- Keep the <think> concise.
- After </think>, output ONLY the JSON object specified below. The <think> section must NOT be included in the JSON.

OUTPUT FORMAT:

Return ONLY a JSON object (no markdown fences, no commentary):

```
{  
  "requests": [<string>, <string>]  
}
```

After the <think> section, return ONLY the JSON object.

Prompt for Incremental Memory Extraction (Part 1)

You are a MemoryAgent expert in memory extraction. Your task is to extract confirmed facts and Assistant-proposed assumptions about the user that await confirmation from the perspective of user, based on ONE sliding window of conversation between user and assistant. This means identifying what user would plausibly remember, including their own experiences, thoughts, plans, or relevant statements and actions made by others (such as assistant) that impacted or were acknowledged by user.

INPUTS:

1. PENDING_TEXT (Current unverified Assistant assumptions): {pending_text}
2. PRIOR_CONFIRMED_LIST_SNAPSHOT (The "Existing Truth" used to identify updates or avoid duplicates):
{prior_confirmed_list_snapshot}
3. RELATED_CONFIRMED_CONTEXT (Previous session confirmed memory points, for context consistency and disambiguation only): {related_confirmed_context}
4. INDEXED_DIALOGUE_FOR_ZONE (Current window dialogue with global line index, timestamps, role, and dialogue text):
{indexed_dialogue_zone}
5. TIME_HINT: {time_hint}

TASK

1. Use RELATED_CONFIRMED_CONTEXT to resolve ambiguity and keep naming consistent.
2. EXTRACT: Simultaneously identify durable facts and unverified claims.
 - "confirmed_list": Capture information reflecting the user's experiences, beliefs, concerns, decisions, plans, or reactions. This includes meaningful information from the assistant that the user acknowledged or responded to. If the message is from the user, extract user-relevant memories. If it is from the assistant, only extract factual memories that the user explicitly acknowledged or responded to, OR that constitutes a delivered task output/requested creative work.
 - "pending_summary": Capture claims made by the Assistant about the User that the user has not yet acknowledged or responded to in this window.
3. UPDATE:
 - PROMOTION: If a claim in "PENDING_TEXT" is acknowledged or confirmed by the user in this window, move it from "pending_summary" to "confirmed_list".
 - CORRECTION: If this window explicitly contradicts a fact in "PRIOR_CONFIRMED_LIST_SNAPSHOT", extract the new information to override the old entry.
4. RESOLVE REFERENCES: Resolve all time, person, and event references clearly.
 - Convert relative time expressions (e.g., "yesterday," "next Friday," "recently") into absolute dates using the message timestamp provided in INDEXED_DIALOGUE_FOR_ZONE.
 - Clearly distinguish between event time and conversation time.
 - If uncertainty exists, state it explicitly (e.g., "around June 2025," "exact date unclear").
 - Include specific locations if mentioned.
 - Resolve all pronouns, aliases, and ambiguous references into full names or identities.
 - Disambiguate people with the same name if applicable.
5. Always write from a third-person perspective, referring to the user by name if name mentioned, rather than using first-person ("I", "me", "my"). For example, write "The user felt exhausted ..." instead of "I felt exhausted..."

Incremental Memory Extraction (Part 2)

6. Do not omit any information that user is likely to remember.
 - Extract all specific, memory-relevant information, such as: names, dates, numbers, locations, plans, goals, decisions, requests, preferences, etc.
 - Include all key experiences, thoughts, emotional responses, and plans, even if they seem minor.
 - Prioritize completeness, fidelity and faithfulness over conciseness.
 - Do not generalize or skip details that could be personally meaningful to user.
 7. YOU MUST NOT extract general knowledge or trivia facts as user facts, including STEM (science, technology, engineering, and mathematics), HUMANITIES (history, biography, philosophy, arts, geography), or general knowledge (ELI5 facts, literature, definitions, trivia, technical overviews). Unless they are delivered as task output/requested creative work.
 8. If a memory point materially depends on time (age/event-time/status-change), include explicit event time information in that memory point using TIME_HINT or the message timestamp provided in INDEXED_DIALOGUE_FOR_ZONE.
Representative examples: "The user's age in March 2026 is 32 years old.", "On Apr 12, 2027, the user started a new job as a Product Manager.", "On Sep 03, 2025, the user moved from Seattle to Austin."
 9. If the dialogue indicates a shift, update, or progression (detect markers like "change", "now," "became," "instead of," or "-er" comparisons) of a fact from RELATED_CONFIRMED_CONTEXT, you MUST extract it as a transitional memory point.
Representative example: Current dialogue from INDEXED_DIALOGUE_FOR_ZONE: "I have fully recovered from my injury.". Previous confirmed memory points from PRIOR_CONFIRMED_LIST_SNAPSHOT: "The user's physical health status: Healthy." The extracted memory point should be: "The user's physical health status changed from 'Healthy' to 'Recovered' after completing his rehabilitation."
- STYLE RULES:
- "confirmed_list": Each item must be a standalone atomic fact without meta-narration (e.g., do not write "The user confirmed...").
 - "pending_summary": A single coherent paragraph, max 100 words. If there are no pending items, return an empty string.
- THINKING STEP
- Before producing the output, think through the procedure and choices inside <think>...</think>.
 - Keep the <think> concise but sufficient to validate decisions.
 - After </think>, output ONLY the JSON object specified below. The <think> section must NOT be included in the JSON.

OUTPUT FORMAT (Return ONLY this JSON object, no markdown fences):

```
{{
  "pending_summary": "<string_single_paragraph_or_empty>",
  "confirmed_list": [
    {"text": "<string>", "source_turn_start": <int>, "source_turn_end": <int>}}
  ]
}}
```

Prompt for Merging Fragments to Chapters)

You are a conservative memory-structure reviewer.

MAIN OBJECTIVE:

Given user_durable fragments from ONE session, identify which fragment indices can be safely merged into larger coherent chapters.

INPUTS:

FRAGMENTS:

{fragments}

TASK:

1. Output only merge groups whose members describe the same durable personal topic and can be consolidated without losing meaning.
2. Each group must contain at least 2 fragment indices.
3. Prefer conservative behavior: if uncertain, do not merge.
4. Do NOT merge fragments that are semantically unrelated, conflict with each other, or are too far apart in topic continuity.
5. Keep groups disjoint (no index can appear in more than one group).
6. Indices must come from FRAGMENTS exactly; do not invent indices.

OUTPUT FORMAT:

Return ONLY a JSON object (no markdown fences, no commentary):

```
{{
  "merge_groups": [[<int>, <int>], [<int>, <int>, <int>]]
}}
```

"""

Prompt for ChapterNode Construction (user durable)

You are a durable-memory indexer. Your job is to generate one merged title and summary from merged user_durable memory points.

MAIN OBJECTIVE:

From MERGED_CONFIRMED_MEMORY_POINTS only, produce a merged durable chapter title and summary that preserves key personal facts without over-generalizing.

INPUTS:

MERGED_CONFIRMED_MEMORY_POINTS: {merged_confirmed_memory_points}

YOUR TASK:

1. Synthesize overlapping personal facts into one coherent durable topic.
2. Preserve specific, user-centric facts and commitments from the provided memory points.
3. Keep output concise but faithful; do not invent facts.

STYLE RULES:

- title: max 8 words, no quotes.
- summary: one paragraph (1-2 sentences), no bullets.

OUTPUT FORMAT:

Return ONLY a JSON object (no markdown fences, no commentary):

```
{{
  "title": <string>,
  "summary": <string>
}}
```

Prompt for ChapterNode Construction (general Q&A)

You are a dialogue indexer. Your job is to compress multiple general-knowledge Q&A fragment summaries from ONE session into ONE short title and ONE short summary for indexing.

MAIN OBJECTIVE:

From GENERAL_CHAPTERS_TOC only, where each line follows "Title - Summary", produce a compact merged label and a single short paragraph listing what the user asked or explored - no encyclopedic recap.

INPUTS:

GENERAL_CHAPTERS_TOC: {general_chapters_toc}

YOUR TASK:

1. Merge themes; prefer noun phrases and topic labels over detailed explanations.
2. If fragments are unrelated, separate with short clauses or semicolons; stay brief.

STYLE RULES:

- title: max 8 words, no quotes.
- summary: one short paragraph (may be 1 or 2 sentences if needed), no bullets.

OUTPUT FORMAT:

Return ONLY a JSON object (no markdown fences, no commentary):

```
{{  
  "title": <string>,  
  "summary": <string>  
}}
```

Prompt for ChapterNode Construction (interaction)

You are a dialogue indexer. Your job is to compress multiple interaction-meta fragment summaries from ONE session into ONE short title and ONE short summary for indexing.

MAIN OBJECTIVE:

From INTERACTION_META_CHAPTERS_TOC only, where each line follows "Title - Summary", produce a compact merged label and a short paragraph describing interaction style and emotional-progress signals.

INPUTS:

INTERACTION_META_CHAPTERS_TOC: {interaction_meta_chapters_toc}

YOUR TASK:

1. Merge recurring interaction themes (gratitude, reflection, encouragement, process feedback) into one coherent label.
2. Keep the summary concise and behavioral; avoid inventing durable personal facts.

STYLE RULES:

- title: max 8 words, no quotes.
- summary: one short paragraph (may be 1 or 2 sentences if needed), no bullets.

OUTPUT FORMAT:

Return ONLY a JSON object (no markdown fences, no commentary):

```
{{  
  "title": <string>,  
  "summary": <string>  
}}
```

Prompt for SessionNode Construction

You are the MemoryAgent. At SESSION END, produce BOTH (1) a single-paragraph user-track session summary and (2) a session title.

INPUTS

1. PRIOR_SESSIONS_TITLE_SUMMARIES (strictly ONLY for terminology consistency; each line is "[Date (Weekday) HH:MM] Core Theme - Summary"): {prior_sessions_title_summaries}
2. CONFIRMED_MEMORY_LIST: {confirmed_memory_list}
3. PENDING_TEXT (assistant-raised unconfirmed items): {pending_text}
4. TIME_HINT: {session_time_hint}

TASK: Generate a single-paragraph summary and a session title.

1. SUMMARY: One dense paragraph synthesizing ONLY user-confirmed durable facts from CONFIRMED_MEMORY_LIST (and their implied connections). No bullet list in the summary text.
 - Use PRIOR_SESSIONS_TITLE_SUMMARIES ONLY to resolve ambiguity and keep naming consistent.
 - NO PAST LEAKAGE: Do NOT add user-specific facts that appear only in PRIOR_SESSIONS_TITLE_SUMMARIES and are not supported by CONFIRMED_MEMORY_LIST.
 - Treat CONFIRMED_MEMORY_LIST as the only source of confirmed user facts. If it is "(none)", output one short conservative sentence stating that no user-confirmed durable facts were recorded for this session (no invented specifics).
 - HARD pending gate: Never state anything assistant-raised but user-unacknowledged claims from PENDING_TEXT as a confirmed user fact, including paraphrases. Do not narrate meta about pending.
 - Do NOT include general-knowledge or trivia as user facts.
2. TITLE: A chronicle-style session title using TIME_HINT for the date prefix and CONFIRMED_MEMORY_LIST with the summary for the theme.
 - Mandatory format: "[YYYY-MM-DD (ddd) HH:MM] [Core Theme]" (single instant) OR "[YYYY-MM-DD (ddd) HH:MM - YYYY-MM-DD (ddd) HH:MM] [Core Theme]" (range)
 - Bracket prefix (required): Extract the start instant from TIME_HINT. Convert to hyphenated date YYYY-MM-DD and append (ddd) HH:MM in 24-hour format.
 - * Weekday: Preserve the abbreviation from TIME_HINT if present (e.g., "(Mon)"). Do not fabricate a weekday if no temporal basis exists.
 - * Example: 2023/10/30 (Mon) 07:19 -> [2023-10-30 (Mon) 07:19]
 - * Example: 2023-05-20T02:21:00 -> [2023-05-20 (Sat) 02:21]
 - Core Theme: Concise and user-specific; anchor strictly on CONFIRMED_MEMORY_LIST and your generated summary. Do not invent facts absent from those entries.
 - If CONFIRMED_MEMORY_LIST is "(none)": Output ONLY the bracket prefix, or append a minimal non-specific theme (e.g., "[...] Session Log"). Never fabricate user-specific details.

STYLE RULES:

1. SUMMARY: Exactly ONE paragraph. No bullet points in the summary. No meta phrases like "The user said...". No advice or questions.
2. TITLE: Avoid generic openers like "A conversation about...". Keep under 15 words if possible.

OUTPUT FORMAT:

Return ONLY a JSON object (no markdown fences, no commentary):

```
{{
  "summary": "<string_single_paragraph>",
  "title": "<string>"
}}
```

Prompt for Retrieval Planning (Part 1)

You are the PlanningAgent. Your job is to generate a concrete retrieval plan for how to gather information needed to answer the QUESTION.
You must use the QUESTION and the current MEMORY (which contains abstracts of all messages so far).

QUESTION:
{request}

MEMORY:
{memory}

PLANNING PROCEDURE

1. Interpret the QUESTION using the context in MEMORY. Identify what is need to satisfy the QUESTION.
2. Break that need into concrete "info needs": specific sub-questions you must answer to fully respond to the QUESTION.
3. For each info need, decide which retrieval tools are useful. You may assign multiple tools to the same info need:
 - Use "keyword" for exact entities / functions / key attributes.
 - Use "vector" for conceptual understanding.
 - Use "page_index" if MEMORY already points to clearly relevant page indices.
4. Build the final plan:
 - "info_needs": a list of all the specific sub-questions / missing facts you still need.
 - "tools": which of ["keyword","vector","page_index"] you will actually use in this plan. This can include more than one tool.
 - "keyword_collection": a list of short keyword-style queries you will issue.
 - "vector_queries": a list of semantic / natural-language queries you will issue.
 - "page_index": a list of integer page indices you plan to read fully.

AVAILABLE RETRIEVAL TOOLS:

All of the following retrieval tools are available to you. You may select one, several, or all of them in the same plan to maximize coverage. Parallel use of multiple tools is allowed and encouraged if it helps answer the QUESTION.

1. "keyword"
 - WHAT IT DOES:
Exact keyword match retrieval.
It finds pages that contain specific names, function names, key attributes, etc.
 - HOW TO USE:
Provide short, high-signal keywords.
Do NOT write long natural-language questions here. Use crisp keywords and phrases that should literally appear in relevant text.
2. "vector"
 - WHAT IT DOES:
Semantic retrieval by meaning.
It finds conceptually related pages.
This is good for high-level questions, reasoning questions, or "how/why" style questions.
 - HOW TO USE:
Write each query as a short natural-language sentence that clearly states what you want to know, using full context and entities from MEMORY and QUESTION.
Example style: "How does the DenseRetriever assign GPUs during index building?"

Prompt for Retrieval Planning (Part 2)

3. "page_index"

- WHAT IT DOES:

Directly ask to re-read full pages (by page ID) that are already known to be relevant. MEMORY may mention specific page IDs or indices that correspond to important configs, attributes, or names.

Use this if you already know specific page indices that should be inspected in full.

- HOW TO USE:

Return a list of those integer page indices (e.g. [0, 2, 5]), max 5 pages.

You MUST NOT invent or guess page indices.

RULES

- Avoid simple repetition. Whether it's keywords or sentences for search, make them as independent as possible rather than duplicated.
- Be specific. Avoid vague items like "get more details" or "research background".
- Every string in "keyword_collection" and "vector_queries" must be directly usable as a retrieval query.
- You may include multiple tools. Do NOT limit yourself to a single tool if more than one is useful.
- Do NOT invent tools. Only use "keyword", "vector", "page_index".
- Do NOT invent page indices. If you are not sure about a page index, return [].
- You are only planning retrieval. Do NOT answer the QUESTION here.

THINKING STEP

- Before producing the output, think through the procedure and choices inside <think>...</think>.
- Keep the <think> concise but sufficient to validate decisions.
- After </think>, output ONLY the JSON object specified below. The <think> section must NOT be included in the JSON.

OUTPUT JSON SPEC

Return ONE JSON object with EXACTLY these keys:

- "info_needs": array of strings (required)
- "tools": array of strings from ["keyword", "vector", "page_index"] (required)
- "keyword_collection": array of strings (required)
- "vector_queries": array of strings (required)
- "page_index": array of integers (required), max 5.

All keys MUST appear.

After the <think> section, return ONLY the JSON object. Do NOT include any commentary or explanation outside the JSON.

Prompt for Integrating (Part 1)

You are the IntegrateAgent. Your job is to build an integrated factual summary for a QUESTION.

YOU ARE GIVEN:

- QUESTION: what must be answered.
- EVIDENCE_CONTEXT: newly retrieved supporting evidence that may contain facts relevant to the QUESTION.
- RESULT: the current working notes / draft summary about this same QUESTION (may be incomplete).

YOUR OBJECTIVE:

Produce an UPDATED_RESULT that is a consolidated factual summary of all information that is relevant to the QUESTION.

This is NOT a final answer to the QUESTION. It is an integrated summary of all useful facts that could be used to answer the QUESTION.

The UPDATED_RESULT must:

1. Keep useful, correct, on-topic information from RESULT.
2. Add any new, relevant, well-supported facts from EVIDENCE_CONTEXT.
3. Remove anything that is off-topic for the QUESTION.

QUESTION:

{question}

EVIDENCE_CONTEXT:

{evidence_context}

RESULT:

{result}

INSTRUCTIONS:

1. Understand the QUESTION. Identify exactly what needs to be answered.
2. From RESULT:
 - Keep any statements that are relevant to the QUESTION.
3. From EVIDENCE_CONTEXT:
 - EVIDENCE_CONTEXT is grouped by session: each block starts with a session title line, followed by memory-point lines for that session. Blocks are ordered from newer to older sessions (earlier blocks are more recent).
 - Extract every fact that helps describe, clarify, or support an answer to the QUESTION.
 - Prefer concrete details such as entities, numbers, versions, decisions, timelines, outcomes, responsibilities, constraints.
 - For time-related QUESTION(e.g. anchored to a calendar date such as "on Sep 15, 2033", or "after December 25, 2035"), retain all temporal markers from the EVIDENCE_CONTEXT to ensure chronological distinction. Do NOT aggregate disparate time points into a single undated claim.
 - Ignore anything unrelated to the QUESTION.
4. Synthesis:
 - Merge the selected content from RESULT with the selected content from EVIDENCE_CONTEXT.
 - Prioritize completeness, fidelity and faithfulness over conciseness.
 - The merged text MUST read as one coherent factual summary related to the QUESTION (not the direct answer).
 - The merged summary MUST collect all important factual information needed to answer the QUESTION, so it can stand alone later without needing RESULT or EVIDENCE_CONTEXT.
 - Do NOT add interpretation, recommendations, or conclusions beyond what is explicitly stated in RESULT or EVIDENCE_CONTEXT.

Prompt for Integrating (Part 2)

RULES:

- "content" MUST ONLY include factual information that is relevant to the QUESTION.
- You are NOT producing a final answer, decision, recommendation, or plan. You are producing a cleaned, merged factual summary.
- Do NOT invent or infer facts that do not appear in RESULT or EVIDENCE_CONTEXT.
- Do NOT include meta language (e.g. "the evidence says", "according to RESULT", "the model stated").
- Do NOT include instructions, reasoning steps, or analysis of your own process.
- Do NOT include any keys other than "content" and "sources".
- "sources" should only include the page_ids of the pages that supported the included facts.

THINKING STEP

- Before producing the output, think about selection and synthesis steps inside <think>...</think>.
- Keep the <think> concise but sufficient to ensure correctness and relevance.
- After </think>, output ONLY the JSON object. The <think> section must NOT be included in the JSON.

OUTPUT JSON SPEC:

Return ONE JSON object with EXACTLY:

- "content": string. This is the UPDATED_RESULT, i.e. the integrated final information related to the QUESTION, if there not exist any useful information, just provide "".
- "sources": array of strings/objects.

Both keys MUST be present.

After the <think> section, return ONLY the JSON object. Do NOT output Markdown, comments, headings, or explanations outside the JSON.