

MemPulse: Memory Pulse-Driven Adaptive Long-Context Reasoning with Fact-based Hypergraph

Wenbiao Tao^{1*}, Yuanfu Wang², Yunshi Lan^{1†}, Chao Yang^{2†}

¹East China Normal University

²Shanghai Artificial Intelligence Laboratory

wbtao@stu.ecnu.edu.cn, wangyuanfu@pjlab.org.cn

yslan@dase.ecnu.edu.cn, yangchao@pjlab.org.cn

Abstract

Memory-augmented long-context reasoning boosts LLMs’ complex task solving via external knowledge, yet current methods suffer from disjointed memory retrieval-induced context drift and aimless iterations causing premature convergence or redundant computation. We present MemPulse, integrating fact-based memory hypergraphs and pulse-driven adaptive iteration. It builds hypergraphs with facts as hyperedges to gather scattered information and sustain coherent context evolution. Its memory pulse engine computes agent confidence and structural novelty to output dynamic control signals, adjusting retrieval range and stopping criteria for logical reasoning with minimal overhead. Evaluated on LoCoMo, MemPulse surpasses mainstream baselines in reasoning accuracy and efficiency.

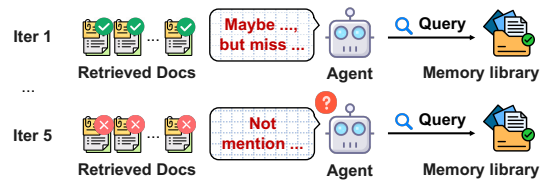
1 Introduction

Large Language Models (LLMs) have revolutionized the landscape of natural language processing, demonstrating remarkable capabilities in reasoning and generation (Hu et al., 2025). To tackle tasks requiring long-term consistency and complex decision-making, these models are increasingly deployed as autonomous agents capable of interacting with external environments and tools (Luo et al., 2025b; Chen et al., 2025). A critical component empowering such agents is agentic memory, which allows them to retain, retrieve, and utilize historical information across extended interaction sessions. By augmenting LLMs with external memory structures, agents can overcome the inherent context window limitations of transformers and maintain coherence over multi-turn dialogues (Zhang et al., 2025). Consequently, memory-augmented reasoning has emerged as a pivotal paradigm, enabling

*This work was done during an internship at Shanghai Artificial Intelligence Laboratory.

†Corresponding authors.

Retrieval-Reasoning Dilemma



MemPulse



Figure 1: Retrieval-reasoning dilemma where isolated retrieval breaks reasoning coherence, while uncontrolled iteration causes memory drift or blind expansion.

agents to perform multi-hop inference by synthesizing fragmented facts stored in their memory banks.

Current memory-augmented reasoning frameworks fall into Retrieval-Augmented Generation (RAG)-based memory reasoning and iterative agentic reasoning. RAG-based memory reasoning (Chhikara et al., 2025; Rasmussen et al., 2025; Li et al., 2025) focuses on elaborately designing memory structures for precise retrieval results. These methods structure memory by constructing hierarchical knowledge graphs, thereby endowing them with representational capacity for complex reasoning. Although graph-based memory delivers rich semantics, the reasoning process lacks coherent memory derivation, resulting in fragmented agentic outputs. Moving beyond isolated memory retrieval, iterative agentic reasoning (Yan et al., 2025) combines multiple rounds of retrieval results with model responses to achieve closed-loop memory reasoning. However, excessive iteration causes memory drift, while insufficient retrieval leads to memory loss, trapping agents in a retrieval-

reasoning dilemma.

Based on the above issues, we rethink long-context reasoning with memory and introduce the following two research questions:

- **(RQ1) How to align dynamic reasoning with coherent memory retrieval?** Traditional retrieval lacks integrative inheritance, causing memory drift and reasoning confusion during complex agentic iterations.
- **(RQ2) How to overcome blind iteration and convergence dilemmas?** Current frameworks lack quantitative evaluation of reasoning inertia, leading to either premature convergence on incomplete answers or aimless search expansion without bridging logical gaps.

To bridge these gaps, we propose MemPulse, a novel framework that unifies Fact-based Memory Hypergraphs with Memory Pulse-Driven Adaptive Iteration. MemPulse first constructs a hypergraph where facts serve as hyperedges linking multiple entities, naturally aggregating dispersed information across disjointed sessions to ensure coherent context inheritance. Central to our approach is the Memory Manifold Evolution mechanism, which employs a coarse-to-fine retrieval strategy to iteratively distill and merge relevant facts, effectively preventing memory drift. To govern the reasoning process, we introduce a Memory Pulse Engine that quantitatively synthesizes the agent’s judgment confidence and the structural novelty of the evolving memory manifold. This engine generates a dynamic “pulse” signal that adaptively modulates the retrieval scope and determines convergence, ensuring the system terminates only when logical gaps are rigorously bridged. By synchronizing structured memory representation with pulse-regulated iteration, MemPulse achieves efficient, self-correcting long-context reasoning without blind expansion or premature stopping.

- We propose a Fact-based Memory Hypergraph with Memory Manifold Evolution to address incoherent retrieval. By modeling facts as hyperedges to aggregate cross-session information and using coarse-to-fine retrieval, it ensures coherent context inheritance and prevents memory drift during dynamic reasoning.
- We introduce a Memory Pulse Engine to govern Adaptive Iterative Reasoning. By convert-

ing confidence and structural novelty into dynamic pulses, it adaptively controls retrieval scope and termination, preventing both blind expansion and premature convergence.

- We evaluate our framework on the LoCoMo dataset, demonstrating its effectiveness and efficiency for long-context reasoning.

2 Related Work

2.1 Retrieval-Augmented Generation

RAG grounds LLMs in external knowledge, mitigating hallucinations (Lewis et al., 2020). While traditional RAG uses dense vector retrieval over flat text, it struggles with multi-hop reasoning due to missing structural connections. Graph-based RAG (Peng et al., 2025) addresses this by leveraging Knowledge Graphs for complex relationships. Recent works (Edge et al., 2024; Guo et al., 2025; Fan et al., 2025; Tao et al., 2026) show it enhances reasoning by integrating graph traversal with semantic search, navigating connected facts. However, existing GraphRAG methods typically employ static query strategies, lacking adaptability to dynamically adjust search depth.

2.2 Memory Systems for Long-context Reasoning

Memory systems are fundamental to long-context reasoning, evolving from simple context buffers to sophisticated long-term stores. Early approaches utilized static retrieval mechanisms, treating memory as a fixed database for single-pass information lookup (Rasmussen et al., 2025). LightMem (Fang et al., 2025) organizes memory into three complementary stages, striking a balance between the performance and efficiency of memory systems. Recent advancements have shifted towards iterative and dynamic memory frameworks, where agents actively plan, retrieve, and reflect over multiple turns. Notable works (Xu et al., 2025; Chhikara et al., 2025) demonstrate the transition to self-evolving memory structures that organize knowledge via dynamic graphs or atomic notes. Furthermore, emerging surveys on Agentic RAG emphasize the necessity of closed-loop feedback, where retrieval strategies adapt based on intermediate reasoning states (Yan et al., 2025). Despite these strides, current systems often struggle to govern the trade-off between retrieval breadth and reasoning convergence, leading to inefficient iterative loops.

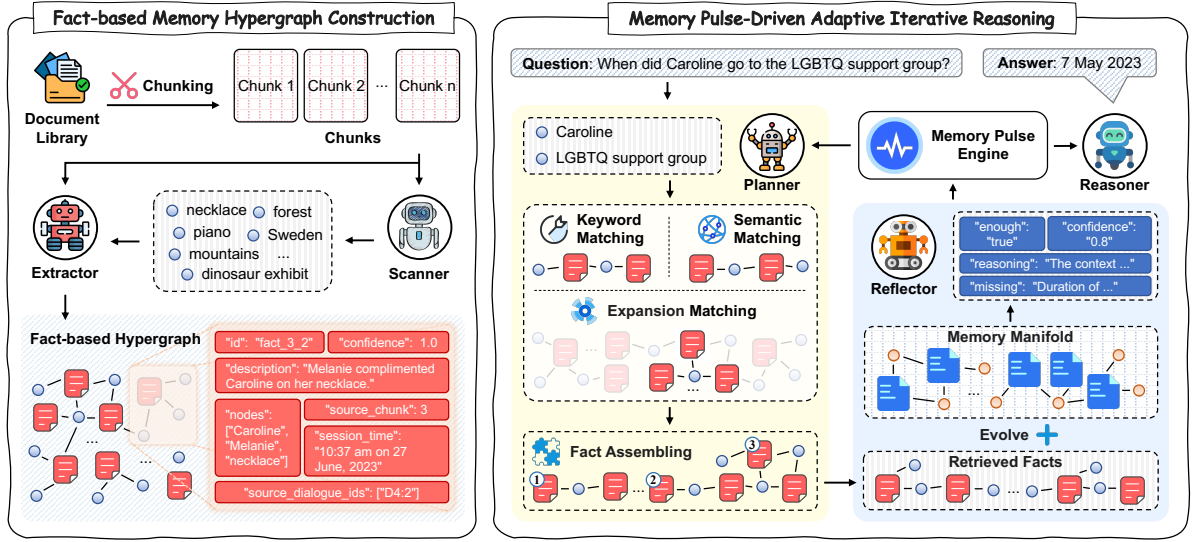


Figure 2: The MemPulse framework constructs a fact-based memory hypergraph from documents (left) and drives adaptive iteration via a Planner-Reflector loop, culminating in final answer deduction by the Reasoner (right).

3 MemPulse

3.1 Framework Overview

As shown in Figure 2, MemPulse consists of two stages, namely Fact-based Memory Hypergraph Construction and Memory Pulse-Driven Adaptive Iterative Reasoning. In order to enable memory to express richer and clearer information, MemPulse abstracts long-context document information into a hypergraph with facts as hyperedges and entities as nodes. In the inference phase, MemPulse guides knowledge retrieval with a memory manifold and drives iterative research based on memory pulses, leading to efficient agentic memory reasoning.

3.2 Fact-based Memory Hypergraph Construction

To represent knowledge in an organized manner, we need to split the document into highly self-contained and fine-grained fact entries, which are connected by shared entities.

Given a document $\mathcal{D}$, MemPulse splits into chunks according to the text structure (such as sessions) or fragments:

$$\mathcal{C} = \{c_i\}_{i=1}^{|\mathcal{C}|} = \text{Chunking}(\mathcal{D}). \quad (1)$$

Since long texts usually involve complex relational links, directly using large language models to construct knowledge graphs can easily overlook certain specific entities and their connections (Lavrinovics et al., 2025). To avoid information omission, we first use the SpaCy NER method as a scanner to

pre-extract entities:

$$\hat{\mathcal{V}}_c = \{\hat{v}_i\}_{i=1}^{|\hat{\mathcal{V}}_c|} = \text{NER}(c), \quad (2)$$

where c is a chunk and $\hat{\mathcal{V}}_c$ denotes the set of identified entities from c . With the hint of named entities, LLMs can extract adequate knowledge from documents in an informed manner. Pursuing a streamlined structure and efficient representation, MemPulse does not extract limited-expression triples, but abstracts facts as hyperedges to construct a fact-based memory hypergraph:

$$\mathcal{H}_c = \{(e_i, \mathcal{V}_i)\}_{i=1}^{|\mathcal{H}_c|} = \text{Extractor}(c, \hat{\mathcal{V}}_c, \text{LLM}), \quad (3)$$

$$\mathcal{G} = \text{Merge}(\{\mathcal{H}_i\}_{i=1}^{|\mathcal{C}|}), \quad (4)$$

where $\mathcal{H}_c$ represents the facts composed of a set of a hyperedge e_i with entity set $\mathcal{V}_i$. Compared with traditional triple-structured knowledge graphs, hypergraphs can directly link factual evidence to multiple entity units, which naturally possesses the characteristic of multi-information aggregation (Luo et al., 2025a).

3.3 Memory Pulse-Driven Adaptive Iterative Reasoning

3.3.1 Iterative Research Pipeline

MemPulse employs agentic reasoning with memory pipeline, where agents driven by LLMs interact through memory to realize reasoning for complex questions. It uses two agents, Planner and Reflector, to iteratively study user questions, and generates research results through the Reasoner agent.

The research process alternates between Planner and Reflector, deeply analyzing the coverage of the problem, the alignment of memory, and the credibility of reasoning. **Planner** aims to distill and align the vast memory bank to the granularity of the question’s involvement. Through the Memory Manifold Evolution Mechanism, which includes Coarse-grained fact retrieval and Fine-grained memory planning, it embeds the matched knowledge into the memory manifold and continuously evolves it over multiple iterations:

$$\mathcal{M}_t = \text{Planner}(q, m, \mathcal{M}_{t-1}, \text{LLM}), \quad (5)$$

where q denotes the user question, m represents the missing information, and $\mathcal{M}_t$ is the memory manifold at the t -th round. To achieve iterative agentic reasoning, a role that intelligently provides retrieval feedback is needed. Given the question and the memory manifold, **Reflector** analyzes the current knowledge, determines whether to iterate, and reflects on missed retrievals:

$$\delta, m = \text{Reflector}(q, \mathcal{M}_t, \text{LLM}), \quad (6)$$

where δ is a binary judgment indicator of whether the iteration should continue.

Powered by the Memory Pulse Engine, MemPulse iterates and stops adaptively. Based on comprehensive memory knowledge and retracing dialogue window, **Reasoner** synthesizes all factual information to generate accurate answers:

$$a = \text{Reasoner}(q, \mathcal{M}, w, \text{LLM}), \quad (7)$$

where a denotes the answer to the user question and w is the window size for retracing the original conversation based on the memory facts.

3.3.2 Memory Manifold Evolution Mechanism

The memory manifold, as a core component of MemPulse, is composed of a subgraph of the memory hypergraph formed by the retrieved facts and carries the working memory of the entire research process, playing a crucial role in the final reasoning. Through iterative memory retrieval, it is distilled from the memory hypergraph and continuously evolves. Planner looks for the appropriate memory in three steps and injects it into the memory manifold, which is called coarse-grained fact retrieval, adaptive coarse screening and Fine-grained memory planning.

Coarse-grained fact retrieval. To alleviate the pressure on LLMs to filter memories, coarse-grained fact retrieval pre-filters all memories.

By extracting the entities of the query with the SpaCy NER method, it first recognizes sparse keyword elements, which are paired with the dense representation of the query to form a complementary retrieval basis:

$$V_q = \{\hat{v}_i\}_{i=1}^{|\hat{V}_q|} = \text{NER}(q_{\text{miss}}), \quad (8)$$

where q_{miss} denotes the query that is the question q at the first turn and the missing information m provided by Reflector at the subsequent turns.

The extracted sparse keyword set V_q is integrated with dense query semantic representations to build complementary retrieval cues. Keyword matching and cosine-similarity-based semantic matching are employed to retrieve information missing from memory. Sorted by semantic similarity, the facts carrying the desired information are obtained to fill the gaps in the memory:

$$F_{\text{gap}}, S_{\text{gap}} = \text{Match}_{\text{gap}}(q, q_{\text{miss}}, V_q). \quad (9)$$

where S_{gap} denotes the fact semantic scores.

In addition, adhering to the memory inheritance based on the memory manifold, MemPulse explores the one-hop of the h nodes with the highest IDF to achieve exploratory facts, ensuring entity rarity spreading:

$$F_{\text{exp}} = \text{Match}_{\text{exp}}(\text{IDF}_{\text{top-h}}(M)), \quad (10)$$

where $\text{Match}_{\text{exp}}(\cdot)$ is the one-hop extension function and $\text{IDF}_{\text{top-h}}(\cdot)$ denotes entity rarity calculation based on IDF.

Adaptive coarse screening. Due to different retrieval principles, the facts from coarse-grained retrieval cannot be filtered uniformly. Therefore, we introduce the adaptive coarse screening strategy. In order to control the retrieval scope, the quantity for coarse screening is calculated based on the Reflector’s confidence and the structural entropy increase of the memory manifold:

$$\lambda = \Pi_{[k, \eta \times k]}(1 + \alpha(1 - s_{\text{conf}}) + (1 - \alpha)(1 - s_{\text{nove}})) \quad (11)$$

where α indicates the confidence weight for extension, $\Pi_{[a, b]}(\cdot)$ is the mapping function from a to b , η denotes the maximum extension factor, and k is the number of facts that the LLM is required to

filter out, as well as the final number of memories retrieved in each round. Additionally, to balance factual diversity, the exploration ratio $\kappa = 1 - s_{\text{nove}}$ is also adaptively determined by the structural entropy increase of memory manifold.

With the support of clever screening strategy, the retrieved facts can be adaptively assembled during the iteration process:

$$F_{\text{coarse}} = \text{Assembly}(F_{\text{gap}}, F_{\text{exp}}, \kappa, \lambda). \quad (12)$$

The fact assembling gradually loads factual evidence according to three levels of priority. The intersection of gap facts and exploratory facts is the highest priority, with gap facts second and exploratory facts as a complement. The detailed algorithm of fact assembling is shown in Appendix B.

Fine-grained memory planning. The coarse filtering eliminated most irrelevant facts, but limited by the ability of rule-based retrieval, it is difficult to accurately identify deep connections between knowledge. To further capture more valuable information, we let an LLM act as the precise retriever to filter out fine-grained memories:

$$F_{\text{fine}} = \text{LLM}(q, M, F_{\text{coarse}}, k). \quad (13)$$

With the planning of fine-grained memories, the memory manifold evolves by merging them:

$$M_t = \text{Merge}(M_{t-1}, F_{\text{fine}}). \quad (14)$$

By virtue of coarse- and fine-grained memory filtering, MemPulse locates the most helpful factual evidence for problem reasoning from the vast memory hypergraph, thus upgrading the memory manifold. Furthermore, due to the two-stage memory retrieval, the LLM only need to select from a streamlined set of candidate facts, which greatly enhances the evolutionary efficiency.

3.3.3 Memory Pulse Engine

Adaptive iterative reasoning requires a signal to determine whether to proceed to the next round of research. Unlike GAM, which directly relies on LLM judgment, MemPulse introduces the memory pulse engine that comprehensively considers model judgment, confidence, and memory manifold novelty to decide whether to iterate.

According to the question and current memory, the LLM determines whether to continue iterating and generates confidence:

$$b, s_{\text{conf}} = \text{LLM}(q, M), \quad (15)$$

where b is the iteration decision and s_{conf} is the confidence of the iteration decision. Furthermore, in order to avoid decision-making mistakes in LLMs, we additionally introduce structural entropy increase to represent the novelty of memory and reflect the information gain brought by iteration:

$$s_{\text{nove}} = \text{Norm}(H(M_t) - H(M_{t-1})), \quad (16)$$

where $H(\cdot)$ represents structural entropy function (Li and Pan, 2016) and $\text{Norm}(\cdot)$ means the normalization function. Based on the above quantitative indicators, iteration scores can be calculated as follows:

$$s_{\text{iter}} = 1 - (s_{\text{conf}} + s_{\text{nove}})/2, \quad (17)$$

where a higher confidence level gets a larger weight, which allows the model’s judgment to be fully respected. Based on the LLM reflection and memory manifold status, the memory pulse engine could decide the final iteration judgment:

$$\delta = b \vee \mathbb{I}(s_{\text{iter}} > t/T). \quad (18)$$

where t represents the t -th turn and T is the maximum number of iterations.

With the memory Pulse engine, Mem Pulse is capable of comprehensively considering intelligent decision-making and information gain to adaptively achieve iterative memory research.

4 Experimental Setup

4.1 Datasets and Baselines

We evaluate our method on LoCoMo (Maharana et al., 2024), a long-context reasoning benchmark, which focuses on conversational question answering and involves four dimensions: Single Hop, Multi Hop, Temporal, and Open Domain. To validate the performance and efficiency of adaptive long-context reasoning with memory, we compare MemPulse with two types of baselines. (1) **Memory-free methods:** Long-LLM slices chunks by the model’s maximum input length and reports the highest output score. The RAG baseline slices documents by 2048 tokens and recalls top-5 to enhance generation. (2) **Memory-based methods:** A-Mem (Xu et al., 2025), Mem0 (Chhikara et al., 2025), MemoryOS (Rasmussen et al., 2025) and LightMem (Fang et al., 2025) construct elaborate memory structures (e.g., graphs) to distill historical information, thereby aiding reasoning. GAM (Yan et al., 2025) collects memory pages and summaries through a memorizer and achieves iterative memory retrieval and inference with a researcher.

Method	Single Hop		Multi Hop		Temporal		Open Domain		F1-avg	BLEU1-avg
	F1	BLEU1	F1	BLEU1	F1	BLEU1	F1	BLEU1		
GPT-4o-mini										
Long-LLM	46.68	37.54	29.23	22.76	25.97	19.42	16.87	13.70	37.31	29.57
RAG	52.45	47.94	27.50	20.13	46.07	40.35	23.23	17.94	44.73	39.40
A-Mem	44.65	37.06	27.02	20.09	45.85	36.67	12.14	12.00	39.65	32.31
Mem0	47.65	38.72	<u>38.72</u>	27.13	48.93	40.51	28.64	21.58	45.10	35.90
MemoryOS	48.62	42.99	35.27	25.22	41.15	30.76	20.02	16.52	42.84	35.54
LightMem	41.79	37.83	29.78	24.80	43.71	39.72	16.89	13.92	38.44	34.35
GAM	<u>57.75</u>	<u>52.10</u>	42.29	34.44	<u>59.45</u>	<u>53.11</u>	<u>33.3</u>	<u>26.97</u>	<u>53.75</u>	<u>47.51</u>
MemPulse	62.85	56.81	38.32	<u>29.83</u>	60.29	56.16	34.27	27.26	56.04	49.89
Qwen2.5-14b-instruct										
Long-LLM	46.05	39.56	32.08	24.46	30.51	24.45	14.89	11.41	38.31	31.89
RAG	47.87	42.79	26.38	19.54	30.78	25.97	14.16	10.52	38.27	33.01
A-Mem	33.75	30.04	22.09	15.28	27.19	22.05	13.49	10.74	28.98	24.47
Mem0	42.58	35.15	31.73	24.82	28.96	26.24	15.03	11.28	36.04	29.91
MemoryOS	46.33	41.62	38.19	29.26	32.24	27.86	20.27	15.94	40.28	34.89
LightMem	34.92	31.22	25.45	19.61	32.03	27.7	15.81	11.81	31.39	27.15
GAM	<u>58.63</u>	<u>52.95</u>	<u>43.13</u>	<u>34.93</u>	<u>50.43</u>	<u>44.62</u>	32.65	27.99	<u>52.46</u>	<u>46.36</u>
MemPulse	61.53	56.15	43.55	35.27	54.67	48.93	<u>24.88</u>	<u>20.54</u>	54.52	48.60

Table 1: Performance results of memory-free methods, memory-based methods and MemPulse with GPT-4o-mini or Qwen2.5-14b-instruct as backbone on LoCoMo.

4.2 Implementation Details

We use GPT-4o-mini and Qwen2.5-14b-instruct (Team, 2024) as the backbones to conduct the experiments. BGE-M3 (Chen et al., 2024) is employed to embed questions and documents. The maximum number of iterations n is 3, the maximum number k of retrieved facts every round is 15, the confidence weight α is 0.6, the maximum extension factor η is 3, the retrieval window size w is 3 and the one-hop number h is 1.

5 Results and Analysis

5.1 Main Results

Table 1 presents main results of MemPulse and baselines on LoCoMo. We can draw the following conclusions: (1) MemPulse has a significant lead over memory-free methods, Long-LLM and RAG. In both F1-avg and BLEU1-avg metrics, MemPulse exceeds them by over 23.68% and 25.12%, respectively. This indicates that the memory mechanism has a huge benefit to long-context reasoning (2) Compared with non-iterative memory systems such as A-Mem, Mem0, MemoryOS, and LightMem, MemPulse obtains better reasoning ability by virtue of the advantages of planning and reflection. It outperforms by more than 10 points on both metrics. (3) Although GAM has similar performance based on iterative reasoning, MemPulse

still maintains an advantage of approximately 2 points relying on the Memory Manifold Evolution Mechanism.

5.2 Efficiency Analysis

MemPulse is an iterative reasoning framework, so we compare it with GAM of the same paradigm in terms of efficiency. In Figure 3, the efficiency results of MemPulse and GAM is demonstrated, including inference time, input token quantity, and output token quantity. Using memory-enhanced iterative inference like GAM, MemPulse leads in performance while cutting inference time and token consumption by about half. This is attributed to coarse- and fine-grained retrieval, which reduces the time to extract valuable facts from massive amounts of information, as well as the Memory Pulse Engine, which allows for iterations to be performed adaptively instead of bewildering multiple rounds of research. For non-iterative frameworks such as Mem0 and A-Mem, the inference latency is at the 10^3 level (Chhikara et al., 2025). These methods are highly efficient but suffer from a significant performance gap.

5.3 Hyperparameter Sensitivity Analysis

We conduct the hyperparameter sensitivity analysis on the maximum number of iterations n , the number of retrievals per round k , the confidence

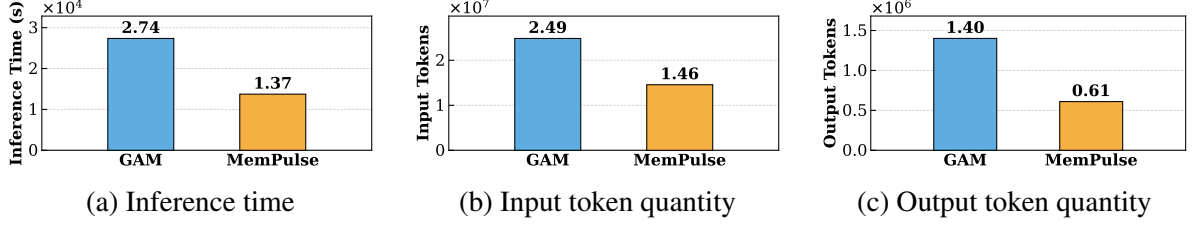


Figure 3: Efficiency analysis: inference time, input token, and output token quantity of GAM and MemPulse.

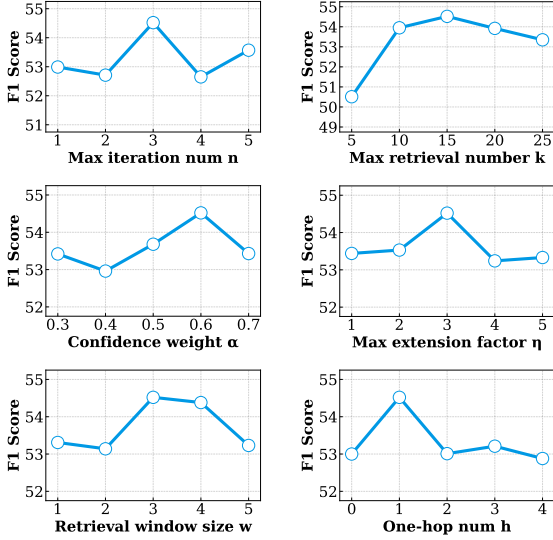


Figure 4: Hyperparameter sensitivity analysis: results on max iteration num n , maximum retrieval number k , confidence weight α , maximum extension factor η , retrieval window size w , and one-hop num h .

weight α , the maximum extension factor η , the retrieval window size w , and the one-hop number h . As shown in Figure 4, more iterative rounds and number of retrievals do not lead to performance benefits but instead cause evidence interference. Balanced model judgment confidence and graph entropy state help iterate for better memories. And restrained coarse sieve expansion as well as one-hop exploration can lead to appropriate retrieval directions and thus improve performance. Throughout the entire hyperparameter sensitivity experiment, MemPulse fluctuates slightly with changes in hyperparameters, which demonstrates its robustness and stability.

5.4 Ablation Study

We perform ablation experiments on LoCoMo to validate five core components. Replacing hypergraphs with ordinary triplets weakens relational modeling and slows knowledge access, hurting overall performance and efficiency. Removing

	F1	BLEU1	Time(s)
MemPulse	54.52	48.60	13.7k
w/o Hypergraph	49.65	43.86	17.4k
w/o Memory Manifold	49.14	42.84	2.4k
Evolution Mechanism			
w/o Memory Pulse Engine	51.73	45.68	2.8k
w/o NER Extraction	52.25	46.26	13.4k
w/o Adaptive Extension	53.45	47.59	20.2k

Table 2: Ablation study: impact of components on performance and efficiency.

memory manifold evolution degenerates the model into question-only one-shot retrieval, failing to fuse valid evidence and yielding sharp accuracy decline. Without the memory pulse engine, iterations become uncontrolled, causing inadequate retrieval or redundant exploration. Eliminating NER extraction leads to missing entities and incomplete fact connections, impairing evidence acquisition. Adaptive extension omission keeps fixed coarse filtering ratio, bringing extra inference cost and degraded results from redundant irrelevant content.

5.5 Iteration Analysis

To prove the robustness of the Memory Pulse Engine, we further statistically analyzed the distribution of confidence, novelty and iteration score under the binary LLM iterative judgment. In Figure 5, confidence is highly consistent with LLM iteration judgment. Judgments that determine an iteration often have a higher level of confidence. When the LLM decides to iterate, novelty is typically high, as the model has identified new information worth exploring. While extremely high or low novelty scores help the LLM determine that further iteration is unnecessary. Combining confidence and novelty, the iteration score distribution and LLM judgment achieve a high degree of correlation. In the interleaved section between 0.2 and 0.4, MemPulse can also achieve higher efficiency through round count penalties.

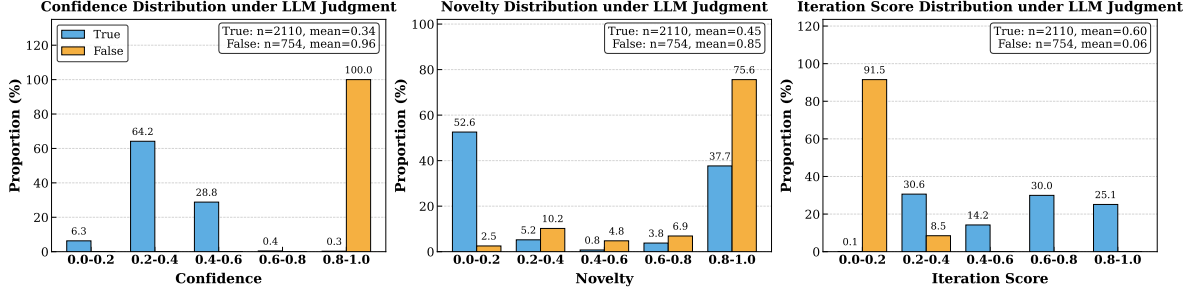


Figure 5: Distributions of confidence, novelty and iteration score by binary LLM iteration judgment.

5.6 Offline Construction Overhead

We further quantify the computational overhead of the offline construction stage, which is a one-time cost incurred before any online inference. Table 3 summarizes the statistics collected from our experimental setup.

Statistical Item	Value
Average Construction Time (s)	1164.9
Average Input Token Quantity	70566.1
Average Output Token Quantity	58831.0

Table 3: Offline construction statistics.

Two critical points merit emphasis. First, this upfront investment is amortized over all subsequent queries, as the constructed index is reused for every inference request. The one-time construction cost pays off substantially through significant inference-time savings, approximately 50% reduction compared to GAM, as reported in Figure 3, which is the primary metric for deployed systems. Second, the reported construction time is measured under a serial, chunk-by-chunk parsing strategy. In practice, the process can be trivially parallelized across independent chunks, reducing the wall-clock time to a negligible level for most deployment scenarios. Therefore, the offline overhead does not pose a practical barrier to real-world adoption.

6 Memory Manifold Statistics

Figure 6 shows the edge and node distribution of the memory manifold. Hypergraph-based memory manifold makes entities closely connected and unit information highly dense. The retrieved facts are almost limited to 35 with entities concentrated between 5 and 20. This indicates that the Memory Manifold Evolution Mechanism effectively controls memory volume for efficient reasoning.

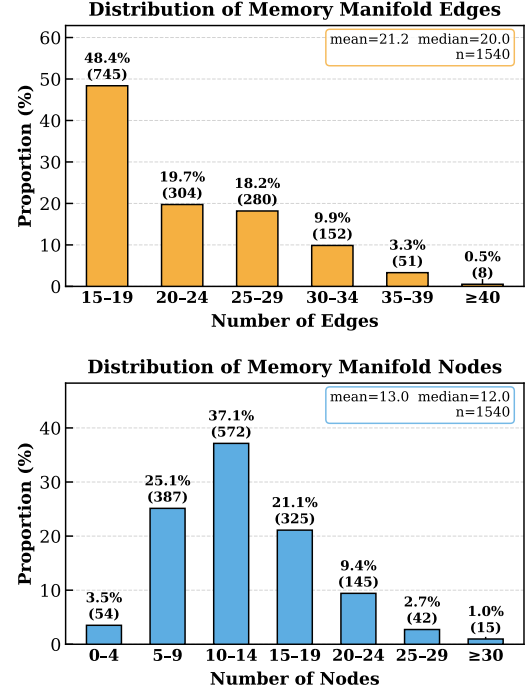


Figure 6: Memory manifold statistics.

6.1 Case Study

Table 4 presents an agentic iteration process case of MemPulse. In the first iteration, MemPulse retrieves some of Caroline’s experiences and involvement based on the question. However, we can see from Reflector’s feedback that even though Caroline’s identity can be inferred through “trans experience” and “transgender conference”, the evidence is still insufficient. Inspired by the retrieved facts, Reflector proposes the missing information for “gender identity”. Based on Reflector’s judgment, MemPulse launches the second round of iteration and identifies the definite “as a transgender woman”. This connects the facts in the first round and leads to the accurate and reliable answer.

Question	What is Caroline’s identity?
Gold answer	Transgender woman
Iteration 1	
Retrieved facts	[fact_10_6]: Caroline’s art expresses her trans experience [fact_4_11]: Caroline is attending a transgender conference this month.
Reflective reasoning	The information provided gives ... While we can infer that Caroline identifies as transgender based on her attendance at a transgender conference and her art expressing her trans experience, the term “identity” in the context of the question could refer to more than just her gender identity. ... Therefore, while we have substantial information about Caroline, we cannot definitively answer the question "What is Caroline’s identity?" without more explicit details.
Missing information	Explicit statement of Caroline’s gender identity or any other specific personal identifier that defines her identity comprehensively.
Iteration 2	
Retrieved facts	[fact_15_8]: Caroline’s art includes a painting that represents her journey as a transgender woman and challenges the binary gender system. ... [fact_2_6]: Caroline felt super powerful giving her talk about her transgender journey .
Reflective reasoning	The retrieved facts provide a comprehensive overview of Caroline’s identity , including her role as a transgender woman, her involvement in LGBTQ activism, her interest in counseling and mental health, and her aspirations related to adoption. These details clearly indicate Caroline’s identity as a transgender woman actively engaged in her community and personal development.
Missing information	No critical information is missing.
Answer	Transgender woman

Table 4: Case study: MemPulse refines iteration via memory manifold and reflector feedback for accurate reasoning.

7 Conclusion

We present MemPulse, addressing memory drift and blind iteration in long-context reasoning. By constructing a Fact-based Memory Hypergraph and employing Memory Manifold Evolution, it ensures robust context inheritance. Additionally, the Memory Pulse Engine enables adaptive reasoning by balancing agent confidence with memory novelty, preventing premature convergence. Experiments on LoCoMo show MemPulse significantly outperforms baselines in accuracy and efficiency, highlighting the value of synchronizing structured memory with dynamic control for complex judgment.

Limitations

While the integration of the NER method enables more complete entity extraction, it inevitably intro-

duces computational overhead. Second, our framework currently lacks a fallback for low-confidence stalemates, where external commonsense knowledge could offer a remedy, a direction we reserve for future work. To further enhance the system’s data processing pipeline, subsequent efforts will involve utilizing robust parsing tools like MinerU to handle diverse multimodal data, enabling seamless integration of multimodal information into MemPulse through parsing and representation.

Acknowledgments

The authors would like to thank the anonymous reviewers for their insightful comments. This work is supported by the National Key Research & Develop Plan (Project No.2023YFF0725100) and Natural Science Foundation of China (Project No. U23A20298).

References

- Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Findings of the association for computational linguistics: ACL 2024*, pages 2318–2335.
- Qinwen Chen, Wenbiao Tao, Zhiwei Zhu, Mingfan Xi, Liangzhong Guo, Yuan Wang, Wei Wang, and Yunshi Lan. 2025. Comrag: retrieval-augmented generation with dynamic vector stores for real-time community question answering in industry. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pages 749–763.
- 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*.
- Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.
- Tianyu Fan, Jingyuan Wang, Xubin Ren, and Chao Huang. 2025. Minirag: Towards extremely simple retrieval-augmented generation. *arXiv preprint arXiv:2501.06713*.
- Jizhan Fang, Xinle Deng, Haoming Xu, Ziyang Jiang, Yuqi Tang, Ziwen Xu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, et al. 2025. Lightmem: Lightweight and efficient memory-augmented generation. *arXiv preprint arXiv:2510.18866*.
- Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. 2025. LightRAG: Simple and fast retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 10746–10761.
- Yuyang Hu, Shichun Liu, Yanwei Yue, Guibin Zhang, Boyang Liu, Fangyi Zhu, Jiahang Lin, Honglin Guo, Shihan Dou, Zhiheng Xi, et al. 2025. Memory in the age of ai agents. *arXiv preprint arXiv:2512.13564*.
- Ernest Lavrinovics, Russa Biswas, Johannes Bjerva, and Katja Hose. 2025. Knowledge graphs, large language models, and hallucinations: An nlp perspective. *Journal of Web Semantics*, 85:100844.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.
- Angsheng Li and Yicheng Pan. 2016. Structural information and dynamical complexity of networks. *IEEE Transactions on Information Theory*, 62(6):3290–3339.
- Rui Li, Zeyu Zhang, Xiaohe Bo, Zihang Tian, Xu Chen, Quanyu Dai, Zhenhua Dong, and Ruiming Tang. 2025. Cam: A constructivist view of agentic memory for llm-based reading comprehension. *arXiv preprint arXiv:2510.05520*.
- Haoran Luo, Guanting Chen, Yandan Zheng, Xiaobao Wu, Yikai Guo, Qika Lin, Yu Feng, Zemin Kuang, Meina Song, Yifan Zhu, et al. 2025a. Hypergraphrag: Retrieval-augmented generation via hypergraph-structured knowledge representation. *arXiv preprint arXiv:2503.21322*.
- Junyu Luo, Weizhi Zhang, Ye Yuan, Yusheng Zhao, Junwei Yang, Yiyang Gu, Bohan Wu, Binqi Chen, Ziyue Qiao, Qingqing Long, et al. 2025b. Large language model agent: A survey on methodology, applications and challenges. *arXiv preprint arXiv:2503.21460*.
- 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.
- Boci Peng, Yun Zhu, Yongchao Liu, Xiaohe Bo, Haizhou Shi, Chuntao Hong, Yan Zhang, and Siliang Tang. 2025. Graph retrieval-augmented generation: A survey. *ACM Transactions on Information Systems*, 44(2):1–52.
- Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. 2025. Zep: a temporal knowledge graph architecture for agent memory. *arXiv preprint arXiv:2501.13956*.
- Wenbiao Tao, Xinyuan Li, Yunshi Lan, and Weining Qian. 2026. Tagrag: tag-guided hierarchical knowledge graph retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 6434–6456.
- Qwen Team. 2024. [Qwen2.5: A party of foundation models](#).
- Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. 2025. A-mem: Agentic memory for llm agents. *arXiv preprint arXiv:2502.12110*.
- BY Yan, Chaofan Li, Hongjin Qian, Shuqi Lu, and Zheng Liu. 2025. General agentic memory via deep research. *arXiv preprint arXiv:2511.18423*.
- Zeyu Zhang, Quanyu Dai, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. 2025. A survey on the memory mechanism of large language model-based agents. *ACM Transactions on Information Systems*, 43(6):1–47.

A Detailed Experimental Settings

A.1 Experimental Environment

Table 5 shows the experimental environment information, including hardware and software.

	Configuration
CPU	INTEL(R) XEON(R) PLATINUM 8558
GPU	NVIDIA H200
RAM	800GB
Operating System	Ubuntu 22.04.4 LTS
CUDA	12.6
Python	3.10.19

Table 5: Experimental environment.

A.2 Dataset Details

We evaluate our method on the LoCoMo benchmark. LoCoMo is a comprehensive long-context benchmark designed to evaluate conversational reasoning over extended narratives. It encompasses four representative question answering dimensions, including single-hop, multi-hop, temporal, and open-domain tasks, which collectively reflect diverse challenges in long-context agent interaction and memory-dependent reasoning. Table 6 shows the conversation statistics and QA benchmark statistics.

Conversation Statistics	Count
Total conversations	10
Average sessions in conversation	27.2
Average turns in session	21.6
QA Benchmark Statistics	Count
Questions in Single Hop	841
Questions in Multi Hop	282
Questions in Temporal	321
Questions in Open Domain	96
Total questions	1540

Table 6: Dataset details.

A.3 Baselines

To validate the performance and efficiency of global question answering, we compare TagRAG with the raw generation of three LLMs with different parameter amounts and the following four RAG-based methods:

- **A-Mem** (Xu et al., 2025): A-Mem is an agentic memory system applying Zettelkasten principles to organize knowledge via structured notes, links, and agent-driven updates.

- **Mem0** (Chhikara et al., 2025): Mem0 is a scalable memory architecture for LLMs that dynamically manages conversational data, optionally using graphs, to boost consistency and reasoning while cutting costs.
- **MemoryOS** (Rasmussen et al., 2025): MemoryOS is a memory layer for agents using Graphiti to merge conversational and business data into an evolving graph for superior long-term context.
- **LightMem** (Fang et al., 2025): LightMem is an efficient memory system inspired by the Atkinson–Shiffrin model, using three stages, sensory, short-term, and long-term, to balance performance with minimal overhead.
- **GAM** (Yan et al., 2025): GAM applies “just-in-time compilation” to agent memory using a lightweight Memorizer and a Researcher to dynamically retrieve and integrate historical data for optimized contexts.

All baselines were aligned with MemPulse’s backbone LLMs (GPT-4o-mini and Qwen2.5-14b-instruct) and generation constraints (e.g. max tokens and temperature).

B Fact Assembling

After retrieving gap facts and exploratory facts, we need to assemble them together according to priority. As shown in Algorithm 1, the intersection of gap and exploratory facts is prioritized because they represent information that is needed for inference and is semantically close. Secondly, for the sake of information completeness, the remaining gap facts have a higher priority compared to exploratory facts.

C Open Domain Performance Analysis

The slight performance decrease on the open-domain subset is a normal task-specific pattern rather than a limitation of the model’s versatility. First, the open-domain questions in LoCoMo focus on scattered, general-knowledge queries with weak structural connections, which is not the primary target scenario of MemPulse. Our framework is designed and optimized for long-context, multi-session conversational reasoning tasks (single-hop, multi-hop, temporal reasoning), where it clearly outperforms all baselines. Second, the open-domain subset is extremely small (only 96 samples, less

Algorithm 1: Fact Assembling

```
1 Input: Gap facts  $F_{\text{gap}}$ , exploratory facts  $F_{\text{exp}}$ , exploration ratio  $\kappa$ , coarse screening quantity  $\lambda$ 
2 Output: Coarse-grained  $F_{\text{coarse}}$ 
3  $F_{\text{intersection}} \leftarrow \text{Top}_{2\lambda}(F_{\text{gap}}) \cap \mathcal{F}_{\text{exp}};$ 
    $S \leftarrow \emptyset;$ 
   // Priority 1: Intersection
4 for  $f \in F_{\text{intersection}}$  do
5   if  $|F_{\text{coarse}}| < \lambda$  and  $f \notin F_{\text{coarse}}$  then
6      $F_{\text{coarse}} \leftarrow F_{\text{coarse}} \cup \{f\}$ 
   // Priority 2: Gap Extension
7  $T_{\text{gap}} \leftarrow \lfloor (\lambda - |F_{\text{coarse}}|) \cdot (1 - \kappa) \rfloor; \text{cnt} \leftarrow 0;$ 
8 for  $f \in F_{\text{gap}}$  do
9   if  $\text{cnt} \geq T_{\text{gap}}$  or  $|F_{\text{coarse}}| \geq \lambda$  then
10    break
11   if  $f \notin F_{\text{coarse}}$  then
12      $F_{\text{coarse}} \leftarrow F_{\text{coarse}} \cup \{f\};$ 
      $\text{cnt} \leftarrow \text{cnt} + 1$ 
   // Priority 3: One-Hop Exploration
13 for  $f \in F_{\text{exp}}$  do
14   if  $|F_{\text{coarse}}| \geq \lambda$  then
15     break
16   if  $f \notin F_{\text{coarse}}$  then
17      $F_{\text{coarse}} \leftarrow F_{\text{coarse}} \cup \{f\}$ 
18 return  $F_{\text{coarse}}$ 
```

than 7% of total data), leading to higher statistical variance. All compared methods show similar performance drops on this subset, confirming it is an inherently challenging split for memory-augmented systems.

We conduct an in-depth case analysis of the reasons why MemPulse performs poorly in open domain tasks. In Table 8, the erroneous reasoning case exposes the deficiency of commonsense concept mapping and inductive reasoning in iterative reflection. All core factual clues were successfully retrieved, including Caroline’s plan to build a children’s library and the confirmation that classic works would be collected in the library. Nevertheless, the reflection module rigidly judged insufficient evidence merely because Dr. Seuss books were not explicitly mentioned in the facts. The model failed to establish logical association between concepts. It could not categorize Dr. Seuss works into classic children’s books, and neglected the inherent inclusion relation that classic chil-

dren’s books belong to the planned library collection. Lacking basic commonsense induction and concept inference capability, the model drew a wrong conclusion despite obtaining valid supporting evidence.

D Stability Analysis

In order to prove the stability and robustness, we further repeat our validation of MemPulse on the LoCoMo dataset three times in Table 7. The fluctuations of F1 and BLEU1 are extremely small, only around 0.1, demonstrating that MemPulse has a stable ability of long context reasoning.

E Prompts

We list the prompts of fact memory extracting, fine-grained memory planning, iterative inference reflecting, and memory-augmented reasoning.

Single Hop		Multi Hop		Temporal		Open Domain		F1-avg	BLEU1-avg
F1	BLEU1	F1	BLEU1	F1	BLEU1	F1	BLEU1		
61.53	56.15	43.55	35.27	54.67	48.93	24.88	20.54	54.52	48.60
61.20	55.68	41.40	33.03	57.90	51.43	25.30	20.50	54.65	48.46
61.64	56.11	42.44	33.87	55.16	49.15	25.56	21.33	54.52	48.42

Table 7: Performance results of MemPulse with Qwen2.5-14b-instruct on LoCoMo.

Question	Would Caroline likely have Dr. Seuss books on her bookshelf?
Gold answer	Yes, since she collects classic children’s books
Iteration 1	
Retrieved facts	[fact_5_5]: Caroline is planning to create a library for her future children. ...
Reflective reasoning	The provided facts do not mention anything about Caroline’s personal preferences for literature, specifically Dr. Seuss books. While we know Caroline plans to create a library for her future children, this does not imply that she herself owns Dr. Seuss books or that these books will be included in her collection. There is no direct evidence linking Caroline to owning or valuing Dr. Seuss books.
Missing information	Information about Caroline’s personal collection of books or her preference for Dr. Seuss books.
Iteration 2	
Retrieved facts	[fact_5_6]: Caroline’s library will include classics, stories from different cultures, and educational books. ...
Reflective reasoning	The provided facts do not mention anything specific about Dr. Seuss books or any preference Caroline might have for children’s literature. While we know Caroline is planning to create a library for her future children, which will include classics, stories from different cultures, and educational books, there is no direct reference to Dr. Seuss or similar children’s authors. Therefore, the information is insufficient to determine if Caroline would likely have Dr. Seuss books on her bookshelf.
Missing information	Information about Caroline’s preferences for children’s literature or possession of Dr. Seuss books.
Iteration 3	
Retrieved facts	[fact_5_5]: Caroline is planning to create a library for her future children. [fact_5_6]: Caroline’s library will include classics, stories from different cultures, and educational books. ...
Reflective reasoning	The retrieved facts provide extensive details about Caroline’s interests, aspirations, and relationships, including her plans to create a library for her future children. However, none of these facts directly mention Dr. Seuss books or indicate any particular preference for children’s literature that might suggest she would have Dr. Seuss books on her bookshelf. While we know Caroline plans to create a library for her future children and intends to include classics, stories from different cultures, and educational books, there is no specific reference to Dr. Seuss or similar children’s authors. Therefore, the information is insufficient to determine whether Caroline would likely have Dr. Seuss books on her bookshelf.
Missing information	Specific information about Caroline’s preferences regarding Dr. Seuss books or any other children’s literature.
Answer	Unlikely

Table 8: Case analysis of incorrect reasoning.

Prompt of fact memory extracting

Extract facts from the following conversation chunk.
{ner_section}

TASK:

Extract FACTS (Complete Statements) from the conversation:

- Extract ALL facts — err strongly on the side of MORE rather than fewer
- Each fact is a complete, self-contained statement
- Include EVERYTHING: actions, events, states, relationships, attributes, preferences, habits, opinions, background details
- Keep all specific details: names, places, dates, numbers, counts, descriptive words
- Extract BOTH main facts AND fine-grained detail facts — do not skip anything even if it seems minor
- DO NOT generalize or abstract away specific information — preserve exact proper nouns from the dialogue
- When any piece of specific information is mentioned, create a fact for it

RULES:

- Extract ALL facts, not just prominent ones
- Resolve pronouns to actual names (use names from the NER list above when available)
- Each fact is self-contained and understandable without context
- Confidence: 1.0 for explicit statements, 0.8-0.9 for implied facts
- ENTITIES: For each fact, list ALL relevant terms with descriptions:
 - * Any named thing: persons, titles, places, organizations, objects, concepts
 - * Include the specific value of attributes when mentioned
 - * Keep entity descriptions brief and grounded in the fact
 - * IMPORTANT: Use FULL NAMES for people — if a nickname/abbreviation is used (e.g., "Mel"), expand it to the full name if known from context (e.g., "Melanie"). If uncertain, use the form that appears most frequently in the conversation.
 - * IMPORTANT: Use the EXACT proper noun from the NER list (or dialogue) as the entity <name> — never a generic substitute

CONVERSATION CHUNK (Index: {chunk_index}):
{chunk_text}

OUTPUT FORMAT:

```
<extraction>
<facts>
<fact>
<id>fact_{chunk_index}_1</id>
<description>Complete statement of what happened</description>
<entities>
<entity>
<name>Entity Name</name>
<description>Brief description of this entity in the context of this fact</description>
</entity>
<entity>
<name>Another Entity</name>
<description>Brief description</description>
</entity>
</entities>
<confidence>0.95</confidence>
<source_dialogue_ids>D1:3,D1:4</source_dialogue_ids>
</fact>
</facts>
</extraction>
```

Extract facts:

Prompt of fine-grained memory planning

You are given a question and a list of candidate facts from memory.
Your task is to select the TOP- $\{top_k\}$ most relevant facts to answer the question.

QUESTION: {question}
{missing_section}
{candidate_facts_text}

TASK:

1. Analyze which facts are most relevant to answering the question
2. Consider:
 - Direct relevance to the question entities and intent
 - Temporal relevance (for time-related questions)
 - Completeness (facts that provide specific details)
3. Select TOP- $\{top_k\}$ fact IDs that are most likely to help answer the question

OUTPUT FORMAT (JSON only):

"selected_fact_ids": ["fact_0_11", "fact_1_5", "fact_2_3"],
"reasoning": "Brief explanation of why these facts were selected"

NOW OUTPUT JSON:

Prompt of iterative inference reflecting

Evaluate whether the retrieved facts contain sufficient information to answer the question.

TASK:

1. Analyze the integration reasoning to understand what information is available
2. Determine if the information is sufficient, complete, and unambiguous to answer the question
3. Output your decision with confidence score

QUESTION:

{question}
{key_entities_section}

RETRIEVED FACTS:

{reasoning_graph_text}
{integration_section}

OUTPUT FORMAT:

<reflection>
<enough>true/false</enough>
<reasoning>Explain why the information is sufficient or what is still missing</reasoning>
<missing>Missing information (phrases without additional analysis and explanation)</missing>
<confidence>0.0-1.0</confidence>
</reflection>

CONFIDENCE GUIDELINES:

- 1.0: All information is explicit, complete, and unambiguous
- 0.8-0.9: Information is mostly complete but may require minor inference
- 0.5-0.7: Information is partial or requires significant inference
- 0.3-0.4: Information is incomplete or ambiguous
- 0.0-0.2: Critical information is missing

Evaluate and output your reflection:

Prompt of memory-augmented reasoning

Based on the dialogue context, answer the following question.

QUESTION:
{question}

DIALOGUE CONTEXT:
{dialogue_context}

CRITICAL OUTPUT FORMAT:

You MUST output in JSON format with two fields:

1. "answer": The direct answer
2. "source_dialogue_ids": A list of dialogue IDs (dia_id) that contain the answer information

ANSWER RULES:

- Factual (Who/What/Where/Which) questions: 1–5 word phrase, use exact words from dialogue
- Time (When) questions: MUST convert relative expressions to absolute time using the session date shown in the dialogue context header (e.g. "SESSION 4 - Dialogue Time: 4 May 2025"):
 - * "yesterday" → "3 May 2025"
 - * "last week" → "the week before 4 May 2025"
 - * "last weekend" → "the weekend before 4 May 2025"
 - * "last year" → "2024" (if session year is 2025)
 - * If no session date is available, keep the relative expression as-is
- Reasoning/opinion (Would/Could/Should): 1 sentence only — core answer + brief qualifier
- How/Why questions: 2 sentences max
- ****CRITICAL FOR LIST QUESTIONS**** (What activities/books/foods/games/places/etc.):
 - * You MUST scan ALL dialogue turns in the context across ALL sessions
 - * Extract EVERY item mentioned, even if scattered across different sessions
 - * Combine them into a comma-separated list
 - * Do NOT omit any items - completeness is critical
 - * Example: If session 1 mentions "basketball", session 3 mentions "swimming", and session 5 mentions "hiking", your answer MUST include all three: "basketball, swimming, hiking"
- NEVER start with "Based on", "According to", "The answer is", or any preamble
- NEVER output full paragraphs or explanations for factual/time questions

SOURCE DIALOGUE IDS RULES:

- The source_dialogue_ids should be a list of dia_ids from the dialogue context
- Format: ["Dsession:turn", ...] (e.g., ["D1:3"], ["D2:5"], ["D3:7"])
- Include ALL dialogue turns that contain key information for your answer
- If the answer comes from a single turn, use a list with one element
- If the answer requires information from multiple turns, include all relevant dia_ids
- ****IMPORTANT****: For questions with multiple candidate turns, choose the turn that BEST MATCHES the question's specific action or intent
 - * Match the verb/action in the question precisely
 - * Match the time context if specified
 - * Choose the most relevant turn, not just any turn mentioning the topic

OUTPUT FORMAT (JSON only, no other text):

"answer": "your short answer here",
"source_dialogue_ids": ["Dsession:turn", ...]

{Examples}

NOW ANSWER (output JSON only):