HMARS: A Hierarchical Multi-Agent Memory System for Long-Context Reasoning

Anonymous ACL submission

Abstract

Long-context reasoning requires models to access, retrieve, and integrate evidence scattered across documents, dialogues, and accumulated interaction histories. Standard retrieval-augmented generation reduces this problem to top-K chunk retrieval, but such passive access can discard relevant evidence before reasoning begins, especially when relevance depends on broader context. We propose HMARS, a hierarchical multi-agent memory system that treats long contexts as managed memory rather than a flat retrieval corpus. Sub-agents maintain grounded access to bounded memory regions, mid-agents manage regional context and provide query-specific coordination, and a frontier model performs final reasoning over retrieved evidence pages. To evaluate this view, we construct two diagnostic benchmarks targeting evidence breadth and context-dependent relevance. Across long-document and multi-turn memory tasks, HMARS achieves the best overall performance against retrieval, reranking, full-context, graph-based, and agentic long-context baselines. Evidence coverage analysis further shows that its gains come from retrieving the required supporting evidence more completely, rather than merely changing the final answer prompt.

1 Introduction

Large language models (Yang et al., 2025; Comanici et al., 2025; Achiam et al., 2023; Singh et al., 2026a) are increasingly expected to reason over long documents, multi-turn dialogues, and accumulated interaction histories. However, simply expanding context windows does not solve long-context reasoning: inference remains expensive, attention over long inputs is unreliable, and relevant evidence is often buried among distractors. Retrieval-augmented generation (RAG) (Jiang et al., 2024; Lewis et al., 2020) remains the default solution (Bai et al., 2024, 2025): it chunks documents, embeds chunks independently, retrieves the top-K, and passes them to a generator. This passive interface is efficient, but assumes relevance is local and statically measurable. The assumption breaks when evidence is scattered across distant regions or becomes relevant only under broader context—for example, an early "$2,500 budget" may be essential to a later question about "housing constraints" despite low lexical overlap. RAG stores context; it does not provide a memory system for managing, accessing, and reasoning over evidence.

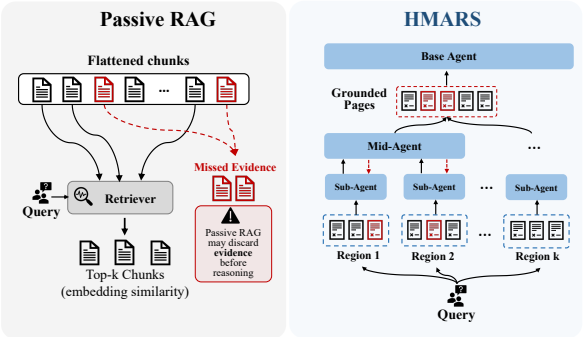

Figure 1: Passive RAG chunks documents, embeds chunks independently, and retrieves top-K evidence, but may miss context-dependent evidence. HMARS treats long inputs as managed memory, where sub-agents assess local regions, mid-agents coordinate shared context, and a frontier model performs final reasoning.

Recent work has begun to move beyond passive retrieval, but still falls short of a memory system for long-context reasoning. Memory-augmented agents (Chhikara et al., 2025; Xu et al., 2026; Zhou et al., 2025; Zhang et al., 2026; Sun et al., 2026) mainly study what to store, update, compress, or forget over long-horizon interactions. Long-context retrieval and reading methods (Jiang et al., 2024; Sarthi et al., 2024; Lee et al., 2024) improve access through larger chunks, summary hierarchies, or page lookup, but remain passive or centralized: some module still decides which parts of the input are allowed to reach the reasoner. What is missing is a memory system that organizes long inputs into local regions, lets each region participate in query-conditioned access, retrieves grounded evidence,

and leaves final reasoning to a stronger model.

We propose HMARS, a Hierarchical Multi-Agent Memory System for long-context reasoning. HMARS treats long documents and interaction histories as managed memory rather than a flat chunk corpus. Lightweight sub-agents retain grounded access to bounded memory regions, mid-agents manage shared regional context and provide query-specific coordination, and a frontier base model performs final reasoning over retrieved evidence. Unlike top-K retrieval, HMARS does not pre-filter away document regions: every sub-agent performs query-conditioned assessment over its local memory, while mid-agent guidance helps retrieve focused, grounded evidence, shown in Figure 1.

To evaluate this memory-system view, we construct two diagnostic benchmarks that stress evidence breadth and context-dependent relevance rather than short sequential hops. The first is derived from LongBench-v2 (Bai et al., 2025), with questions filtered or generated to require evidence across distant chunks and low RAG recall; the second is a cumulative interaction benchmark inspired by RealMem (Bian et al., 2026), where key facts are scattered across multi-turn interactions and become relevant only under later context. Across these settings, HMARS improves most where passive top-K retrieval and centralized memory access fail, supporting our core claim: long-context reasoning should be framed not only as retrieving more chunks, but as managing, accessing, retrieving, and reasoning over memory.

Our contributions are threefold. First, we build harder diagnostic benchmarks for long-context reasoning and multi-turn interaction memory, where questions require broader evidence coverage across distant chunks, regions, or turns. Second, we frame these challenges as failures of passive chunk-based memory access, especially under evidence breadth and context-dependent relevance. Third, we introduce HMARS, a hierarchical multi-agent memory system that separates local memory access, mid-level coordination, grounded evidence retrieval, and frontier-model reasoning.

2 Related Work

2.1 Long-Context Retrieval and Hierarchical Reading

Retrieval-augmented generation remains the dominant interface for long-context QA (Singh et al., 2026b; Jin et al., 2025; Gupta et al., 2024). Recent work improves this pipeline through adaptive retrieval and critique (Asai et al., 2024), retrieval correction, larger retrieval units (Jiang et al., 2024), hierarchical summary trees (Sarthi et al., 2024), or gist memory with page lookup (Lee et al., 2024). These methods improve access to long context, but still rely on passive or centralized selection: a retriever selects chunks, a summary tree is searched, or a single reader controls lookup. HMARS instead treats long context as an agent-managed memory hierarchy, where local regions participate in query-conditioned assessment and return grounded pages without early top-K exclusion.

2.2 Multi-Agent Long-Context Processing

Multi-agent frameworks have also been explored for long-context reasoning (Wang and Chen, 2025; Jiang et al., 2025; Zhao et al., 2024). Chain-of-Agents (Zhang et al., 2024), for example, assigns chunks to sequential worker agents that pass compressed communication units before a manager produces the final answer. These methods show that agent collaboration can reduce input bottlenecks, but they still treat long-context processing primarily as *sequential reading*: each query triggers a pass over the input, and information is propagated through compressed messages. HMARS instead treats the input as a persistent memory system: sub-agents retain grounded page access to local regions, mid-agents provide shared regional context and query-specific coordination, and the base model reasons over retrieved evidence.

2.3 Memory-Augmented LLM Agents

A growing line of work studies explicit memory systems for LLM agents. Mem0 (Chhikara et al., 2025) extracts and retrieves salient conversational memories; A-Mem (Xu et al., 2026) organizes memories as dynamically linked notes; MEM1 (Zhou et al., 2025) jointly learns memory consolidation and reasoning; HiMem (Zhang et al., 2026) builds hierarchical Episode and Note memories; and reflective or learned memory managers study how agents should update, compress, or delete memories over time (Tan et al., 2025; Yan et al., 2025; Yu et al., 2025). While these works establish memory as a central substrate for LLM agents, they mainly focus on the lifecycle of persistent memories: what to store, update, consolidate, or forget. HMARS addresses the complementary problem of accessing a given long context as memory, so that local regions can be queried, grounded evidence

can be retrieved, and no region is discarded before query-conditioned assessment.

3 Method

We present HMARS, a Hierarchical Multi-Agent Memory System for long-context reasoning over documents and interaction histories. Rather than treating a long input as a flat corpus of retrievable chunks, HMARS organizes it as a hierarchy of local memory regions. Sub-agents retain grounded page-level access to bounded regions, mid-agents coordinate these local memories through shared regional context and query-specific guidance, and a frontier base model performs final reasoning over the retrieved evidence pages.

3.1 Problem Setup

Given a long document or interaction history $\mathcal{D}$ and a query q, the goal is to produce an answer a using a compact evidence context $\mathcal{C}(q) \subset \mathcal{D}$. We focus on settings where the answer cannot be reliably recovered from a small number of independently retrieved chunks: evidence may be scattered across distant document regions, or a passage may become relevant only under information stated elsewhere. Thus, the system must avoid discarding regions through an early top-K filter, while still preventing the frontier model from reading the entire context.

3.2 Overview

Given a document and a query, HMARS follows the two-phase workflow in Figure 2. In the offline *indexing phase*, the document is tokenized and partitioned into local regions, each managed by a sub-agent with page-level access. Sub-agents are grouped under mid-agents, which aggregate local memory cards into shared regional memory. In the online *query phase*, the query is routed through the hierarchy to identify relevant grounded pages, which are assembled into the evidence context for final answer synthesis by the base agent.

A key design principle is *asymmetric model allocation*. We instantiate sub-agents with a lightweight model $\mathcal{L}_{\text{sub}}$, mid-agents with a stronger coordination model $\mathcal{L}_{\text{mid}}$, and the base agent with a frontier model $\mathcal{L}_{\text{base}}$:

$$\mathcal{L}_{\text{sub}} \ll \mathcal{L}_{\text{mid}} \ll \mathcal{L}_{\text{base}}. \quad (1)$$

This separates local memory management, regional coordination, and final reasoning: small models manage local pages, mid-level models share regional context, and the frontier model reasons only over retrieved evidence rather than the full document.

Prompt 1: Local Memory Card Creation

You are a summarization agent for a long document section.

Below are the pages (numbered) in your section. Write a concise summary (3–6 sentences) that covers the main topics, entities, events, and any distinctive facts/numbers mentioned. The summary will be read by a higher- level agent that routes queries, so be specific about what THIS section contains versus general background.

Pages:

```
{pages_text}
```

Return ONLY the summary text, no prose preamble.

Indexing phase. The document is first partitioned into bounded local regions. Each region is assigned to a sub-agent s. Within each sub-agent, the assigned tokens are further divided into fixed-size pages $\{p_1, \ldots, p_n\}$. Each page stores its original token offsets, enabling selected pages to be reconstructed in document order.

Each sub-agent then creates a *local memory card*

$$c = (\text{summary}, \text{topics}, \text{entities}, \text{page_index}), \quad (2)$$

with Prompt 1, where summary describes the local region, topics and entities expose salient content, and page_index preserves page-level grounding. The memory card is not used as final evidence; rather, it exposes what the sub-agent knows to its mid-agent.

Sub-agents are grouped under mid-agents. A mid-agent m_j receives the memory cards $\{c_1^j, ..., c_k^j\}$ from its assigned sub-agent group $\mathcal{S}_j = \{s_1^j, ..., s_k^j\}$ and constructs a *shared regional memory* with Prompt 2:

$$\mathcal{R}_j = m_j(c_1^j, ..., c_k^j) \quad (3)$$
$$= (u_j, \text{submap}_j, \text{crossref}_j), \quad (4)$$

where u_j is a region-level summary, submap_j maps each sub-agent to its main topics and entities, and crossref_j records possible cross-sub-agent dependencies such as shared entities, related events, temporal updates, or repeated constraints. This shared regional memory is the mechanism by which otherwise isolated sub-agents exchange information through the mid-agent.

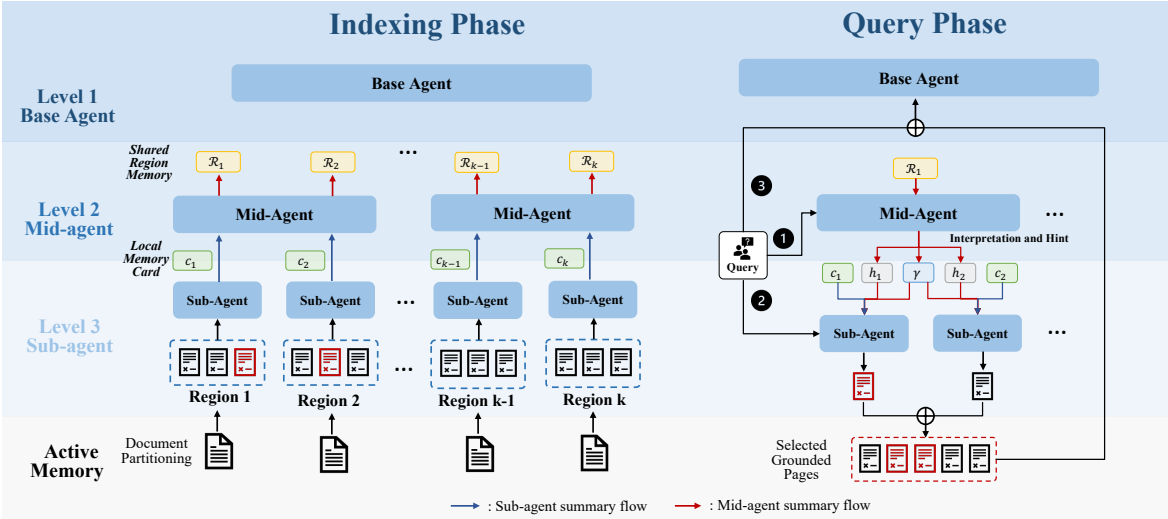

Figure 2: Overview of HMARS. **Indexing phase:** The document is partitioned into local page-level regions; sub-agents build local memory cards, which mid-agents aggregate into shared regional memory. **Query phase:** Mid-agents broadcast query interpretations and sub-agent-specific hints; all sub-agents select relevant pages in parallel, and the base agent synthesizes the final answer from the ordered grounded evidence.

Prompt 2: Region Summary

You manage `{n}` sub-sections of a long document. Below are the summaries produced by each sub-section.

`{sub_summaries}`

Write a region-level summary in two parts, as valid JSON:

`{Json_Format}`

Each sub_map entry lists the primary topics/entities in that sub-section (3–6 topics, 3–8 entities). Return ONLY the JSON object.

Query phase. Given a query q, each mid-agent first interprets the query against its shared regional memory. Instead of producing a hard routing decision, the mid-agent produces a region-level query interpretation γ_j and sub-agent-specific hints with Prompt 5 in Appendix F:

$$(\gamma_j, \{h_i^j\}_{s_i^j \in \mathcal{S}_j}) = \mathcal{L}_{\text{mid}}\big(\text{BroadcastPrompt}(q, \mathcal{R}_j)\big). \tag{5}$$

The interpretation γ_j identifies how the query may relate to the region, including relevant entities, constraints, temporal cues, and possible cross-sub-agent dependencies. Each hint h_i^j tells sub-agent s_i^j what to look for in its own local pages. For sub-agents that are not strongly implicated by the regional memory, the mid-agent still provides a default query interpretation rather than excluding them from participation.

Then, all sub-agents perform local page selection in parallel. Each sub-agent receives the query, its own local memory card, and the mid-agent broadcast with Prompt 6 in Appendix F:

$$\mathcal{L}_{\text{sub}}\big(\text{SelectPrompt}(q, c_i^j, \underbrace{\gamma_j, h_i^j}_{\text{from } m_j}, \underbrace{p_1^i, \ldots, p_{n_i}^i}_{\text{pages for } s_i})\big), \tag{6}$$

The output is a set of page IDs with optional confidence scores and rationales. Because every sub-agent performs this local assessment, no document region is removed by an early top-K filter. At the same time, selection is not independent across sub-agents: each local decision is conditioned on the shared regional interpretation produced by its mid-agent.

Finally, selected pages are deduplicated, sorted by global token offset, and mapped back to their original text spans. The base agent receives the resulting grounded context $\mathcal{C}(q)$ and generates the answer with Prompt 3 in Appendix F:

$$a = \mathcal{L}_{\text{base}}\big(\text{AnswerPrompt}(q, \mathcal{C}(q))\big). \tag{7}$$

This hierarchy separates three roles: sub-agents manage local page memory, mid-agents provide shared regional context and query-specific information sharing, and the base agent performs final reasoning over grounded evidence.

4 Experiments

We evaluate HMARS on two complementary benchmarks designed to expose the structural limitations of top-K retrieval: a long-document understanding benchmark requiring cross-chunk synthe-

Table 1: Long-document benchmark statistics. All questions require evidence from at least four chunks and have RAG recall ≤ 0.5.

Reasoning Type	Count	Avg. Supp. Chunks
Synthesis	73	8.9
Comparison	47	8.4
Causal	28	8.7
Cross-reference	28	8.2
Inference	19	8.5
Total	**195**	**8.6**

sis, and a multi-turn memory benchmark requiring integration of scattered conversational facts.

4.1 Benchmarks

We evaluate HMARS on two diagnostic benchmarks designed to isolate the failure modes of passive top-K access: *evidence breadth*, where answers require integrating many scattered pieces of evidence, and *context-dependent relevance*, where a passage becomes useful only under information stated elsewhere. The first benchmark targets long-document reasoning over distributed evidence; the second targets cumulative memory over multi-turn interactions. Dataset construction details, quality control, and examples are provided in Appendix A.

4.1.1 Long-Document Evidence-Breadth Benchmark

We derive a long-document benchmark from LongBench-v2, selecting 29 documents of 20K–120K tokens across legal, scientific, literary, and game-theory domains. For each document, we generate and filter questions whose answers require evidence from multiple distant chunks rather than a single localized span. Each retained question must require at least four supporting chunks and have embedding-based RAG recall at most 0.5 when retrieving K chunks, where K equals the number of supporting chunks.

The final benchmark contains 195 questions, with an average of 8.6 supporting chunks per question and mean RAG recall of 0.338. This makes the benchmark diagnostic of cases where top-K retrieval fails to expose the full evidence set. Table 1 summarizes the reasoning-type distribution.

4.1.2 Cumulative Interaction Memory Benchmark

To evaluate context-dependent relevance, we construct a multi-turn memory benchmark inspired by RealMem. It contains 120 dialogues across 30 realistic scenarios, including travel planning, apartment search, debugging, and task coordination. Each dialogue contains 24–49 turns and 5–9 key facts scattered throughout the interaction, such as names, numbers, dates, preferences, constraints, and user updates.

Each sample ends with a question that requires recalling and integrating facts from across the conversation. The benchmark covers five memory subtypes: requirement accumulation, progressive updates, long-range dependency, cross-turn integration, and preference tracking. Unlike standard retrieval tasks, relevant turns often have low lexical overlap with the final question and become important only under later context.

4.2 Experimental Setup

Models. HMARS uses an asymmetric model hierarchy: Qwen3-0.6B (Yang et al., 2025) builds sub-agent local memory cards, Qwen3-8B performs mid-agent coordination and local page selection, and GPT-5.1 (Singh et al., 2026a) performs final answer synthesis. Although each sub-agent owns a local memory region, its query-time page selection is executed by the Qwen3-8B selection model conditioned on the sub-agent memory card and mid-agent hint. The Qwen models are served with vLLM. Unless otherwise stated, all inference uses temperature 0.1. Additional decoding parameters and prompts are provided in Appendix B.

Baselines. We compare HMARS against five families of baselines. First, we use dense RAG with Qwen3-Embedding-0.6B and the same frontier answer model as HMARS, varying both retrieval budget and chunk size. Second, we evaluate Dense Rerank, which retrieves a large candidate pool and reranks candidates with Qwen3-8B before answer synthesis. Third, we include a Full-Context LLM baseline that feeds the entire document, up to the model context limit, directly to the frontier model. Fourth, we compare against GraphRAG, which performs local search over an LLM-extracted entity graph. Finally, we include agentic long-context baselines: Chain-of-Agents (Zhang et al., 2024), which sequentially passes compressed communication units across chunk-level workers, and MemAgent (Yu et al., 2025), which maintains a sequential memory state while reading long inputs. Implementation details are provided in Appendix B.

Evaluation. Following LongBench-v2, we use GPT-based judging: for each question, gold answer,

Table 2: Main comparison on the long-document understanding and multi-turn memory benchmarks (accuracy in %). RAG is evaluated with both 1K and 4K retrieval chunks and multiple retrieval budgets. Dense Rerank retrieves N candidates with the same embedder, scores them with Qwen3-8B, and passes the top K chunks to the answer model. GraphRAG uses local search over an LLM-extracted entity graph with community summaries. Chain-of-Agents and MemAgent represent agentic long-context processing and sequential memory baselines. HMARS uses page size $P = 1024$.

Method	**Understanding (195)**					**Multi-Turn**	**Overall**
	Syn. (73)	Cmp. (47)	Cau. (28)	Xref. (28)	Inf. (19)	Acc. (120)	Acc.
RAG ($K = 10$, 1K)	65.8	53.2	60.7	71.4	68.4	60.8	62.2
RAG ($K = 20$, 1K)	82.2	66.0	60.7	82.1	**94.7**	81.7	78.4
RAG ($K = 50$, 1K)	75.4	57.5	68.0	75.1	79.0	86.7	76.5
RAG ($K = 3$, 4K)	56.2	29.8	42.9	53.6	47.4	87.5	62.2
RAG ($K = 5$, 4K)	56.1	29.8	53.5	57.0	57.8	81.7	61.9
Full-Context LLM	74.0	59.6	53.6	75.0	84.2	80.0	73.0
Dense Rerank ($N = 50$)	76.7	63.8	67.9	71.4	84.2	84.2	76.8
Dense Rerank ($N = 100$)	78.1	66.0	64.3	78.6	89.5	89.2	80.0
GraphRAG (Edge et al., 2024)	64.4	57.4	64.3	67.9	57.9	84.0	70.7
CoA-7B (Zhang et al., 2024)	5.5	8.5	7.1	10.8	0.0	40.0	19.6
MemAgent-14B (Yu et al., 2025)	2.6	2.1	7.2	3.6	0.0	65.8	27.0
HMARS (ours)	**83.0**	**66.9**	**70.8**	**91.7**	92.0	**90.8**	**83.8**

and model prediction, the judge assigns a binary correctness score with a brief justification. We report accuracy as the primary metric. For efficiency, we additionally report retrieved prompt tokens, and estimated FLOPs per query.

4.3 Main Results

Table 2 compares HMARS with dense retrieval, reranking, full-context prompting, graph-based retrieval, and agentic long-context memory baselines. HMARS achieves the best overall performance and obtains the highest accuracy on the multi-turn benchmark. On long-document understanding, it is also the strongest method in most reasoning categories, including synthesis, comparison, causal reasoning, and cross-reference. The only category where it does not lead is inference, where a high-budget RAG setting performs best. This pattern supports our central claim: when questions require broad evidence coverage or context-dependent memory access, organizing the input as managed local memory is more effective than treating it as a flat retrieval corpus.

The baselines show that the gains are not simply due to retrieving more text. Increasing RAG's retrieval budget helps in some cases, but performance remains sensitive to both K and chunk size: coarse 4K chunks hurt long-document understanding, while very large top-K contexts introduce noise. Dense Rerank is the strongest non-HMARS baseline, confirming the value of small-model relevance scoring, but it still ranks a flat candidate pool without local memory ownership or mid-level coordination. Full-context prompting also underperforms HMARS, indicating that exposing all tokens to a frontier model is not a substitute for structured memory access. Chain-of-Agents and MemAgent underperform in our diagnostic setting, suggesting that sequential compression can lose fine-grained evidence when answers require broad, non-adjacent support; we discuss this failure mode in Appendix E. Together, these results suggest that HMARS's advantage comes from a hierarchical memory system: local regions assess their own relevance, mid-agents share regional context, and the frontier model reasons over grounded pages.

4.4 Evidence Coverage Analysis

To verify that HMARS's gains come from improved evidence access rather than final-answer prompting, we measure supporting-chunk recall on the long-document benchmark. For each question, we compute the fraction of annotated supporting chunks covered by the evidence passed to the answer model. This isolates the memory-access stage from the final synthesis stage.

HMARS improves evidence access. Table 3 shows that HMARS covers nearly all annotated supporting chunks before answer generation, while RAG retrieves only a partial evidence set even when its retrieval budget or chunk size is increased. This explains the main-result trend: HMARS is not simply benefiting from a different answer prompt, but from a better memory-access mecha-

Table 3: Evidence coverage comparison on the long-document benchmark.

Method	Supp. Recall	Prompt Tok.	FLOPs/query
RAG ($K=5$, 4K)	0.571	~20K	$\sim 8.6\times 10^{16}$
RAG ($K=10$, 1K)	0.331	~11K	$\sim 4.4\times 10^{16}$
RAG ($K=20$, 1K)	0.572	~20.4K	$\sim 8.4\times 10^{16}$
HMARS	**0.994**	~13K	$\sim 5.2\times 10^{16}$

nism. By allowing every local region to perform query-conditioned assessment, HMARS avoids the early evidence loss caused by top-K filtering.

HMARS improves the recall–cost tradeoff. The coverage gain does not come from sending the entire document to the frontier model. Compared with high-budget RAG variants, HMARS achieves much higher supporting-chunk recall while passing a smaller and more focused context to the answer model. This is the intended role of the hierarchy: lightweight sub-agents provide full-context consideration, and mid-agent scoring compresses the resulting evidence set into grounded pages for final synthesis. In contrast, increasing RAG's K or chunk size exposes more text but still misses substantial evidence and eventually introduces noise. Thus, HMARS shifts computation from frontier-model reading to structured memory management, yielding a stronger recall–cost operating point.

4.5 Ablation on Coverage and Coordination

We first test the two architectural mechanisms most central to HMARS: (1) whether every local region receives query-conditioned assessment, and (2) whether the mid-agent layer is needed for sharing regional context. Figure 3 compares the anchor system against two variants: `dispatched`, where only mid-agent-selected sub-agents participate, and `no-mid`, where the mid-agent layer is removed.

Full-fanout provides coverage. Restricting selection to only dispatched sub-agents drops accuracy from 0.838 to 0.762 overall. The drop is consistent on both benchmarks, showing that summary-level dispatch alone cannot reliably decide which local regions matter. A region that appears irrelevant in a mid-agent summary may still contain pages that become relevant only after the query is paired with the local content. This supports the core design of HMARS: every local memory region should receive query-conditioned assessment before evidence is discarded.

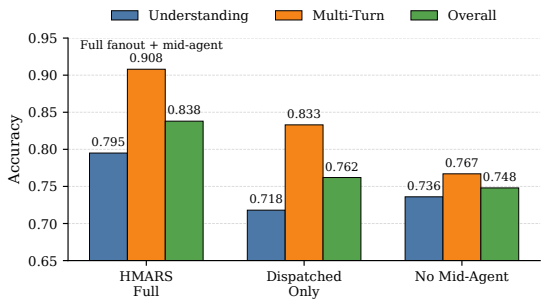

Figure 3: Architectural ablations for HMARS.

Mid-agents provide shared regional context. Removing the mid-agent layer drops overall accuracy from 0.838 to 0.748, with the largest degradation on multi-turn memory. This indicates that sub-agents should not operate as isolated local readers. The mid-agent aggregates sub-agent memories into a shared regional view and broadcasts query-specific guidance, helping local agents interpret their pages under broader context. Thus, full fanout and mid-agent coordination are complementary: full fanout ensures coverage, while mid-agents improve how local memories are interpreted.

4.6 Ablation on Memory Granularity

We next study how sensitive HMARS is to memory granularity: the page size P, which controls the evidence unit returned to base model, and the sub-agent capacity C_{sub}, which controls how much context each sub-agent owns. Figure 4 shows both sweeps.

Page size is not the main driver. Changing P across $\{512, 1024, 2048\}$ yields similar performance, with all overall scores within a narrow range. Smaller pages provide finer-grained grounded evidence, while larger pages preserve more local context and reduce page-scoring overhead. The small variation suggests HMARS does not depend on a finely tuned page size; its gains mainly come from the memory-management structure rather than a particular evidence granularity.

Sub-agent capacity controls the memory partition. Changing C_{sub} has a clearer effect. A smaller capacity fragments the document into too many local memories, while a larger capacity makes each local memory coarser and hurts multi-turn memory. The default $16K$ capacity provides a balanced partition, preserving enough local context within each sub-agent while keeping regional coordination manageable.

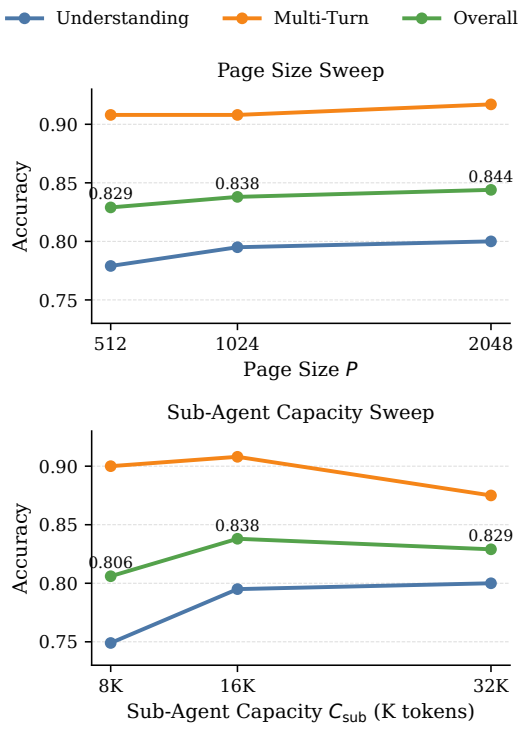

Figure 4: Granularity ablations for HMARS. Page size P controls the evidence unit returned to the base model, while sub-agent capacity C_{sub} controls the size of each local memory region.

4.7 Ablation on Model Allocation

Finally, we test whether local memory management requires a larger sub-agent model, results are shown in Table 4. Replacing the 0.6B sub-agent model with the 8B model yields only a small change on long-document understanding and reduces multi-turn accuracy. Overall accuracy changes from 0.838 to 0.832, suggesting that scaling the sub-agent model does not provide a reliable gain.

Table 4: Sub-agent model size ablation.

Sub-agent Model	UND	MT	Overall
Qwen3-0.6B	0.795	**0.908**	**0.838**
Qwen3-8B	**0.800**	0.883	0.832

This supports the asymmetric allocation used in HMARS: lightweight models are sufficient for local memory summarization and page-level management, while stronger models are better reserved for mid-level coordination and final synthesis.

4.8 Efficiency Analysis

Figure 5 compares representative methods by overall accuracy and estimated per-query FLOPs. HMARS achieves the best overall accuracy with roughly 5.2×10^{16} FLOPs per query, only mod-

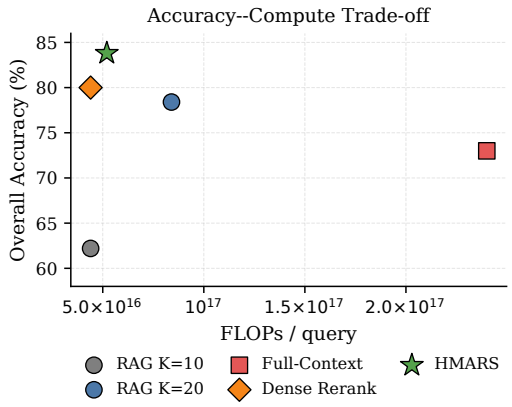

Figure 5: Accuracy–compute trade-off on the full evaluation set.

estly above RAG ($K = 10$) and Dense Rerank (4.4×10^{16}), but well below high-budget RAG and Full-Context prompting. This efficiency comes from asymmetric model allocation: local and mid-level models perform memory access over the full input, while the frontier model reasons only over selected grounded pages.

The comparison shows that more compute does not necessarily lead to better long-context reasoning. Full-Context prompting uses the largest budget (2.4×10^{17} FLOPs) but underperforms HMARS; RAG with a larger retrieval budget reaches 8.4×10^{16} FLOPs while still trailing. Dense Rerank is efficient and strong, but remains a flat candidate-ranking pipeline. Overall, HMARS offers a favorable accuracy–compute trade-off by shifting work from frontier-model reading to structured memory access.

5 Conclusion

We introduced HMARS, a hierarchical multi-agent memory system for long-context reasoning over documents and interaction histories. Rather than treating long inputs as flat retrieval corpora, HMARS organizes them into local memory regions, uses mid-agents to coordinate query-specific regional context, and reserves the frontier model for reasoning over grounded evidence. Across diagnostic benchmarks targeting evidence breadth and context-dependent relevance, HMARS improves over retrieval, reranking, full-context, graph-based, and agentic baselines. These results suggest that long-context reasoning should be framed not only as retrieving more chunks or extending context windows, but as building memory systems that access, coordinate, and ground evidence across the input.

Limitations

HMARS provides full-context consideration, but not guaranteed evidence recall: local agents can still miss relevant pages or select noisy evidence. The current implementation also relies on prompted LLM agents rather than learned memory policies, making performance sensitive to prompts, model choice, and decoding settings. While the asymmetric hierarchy reduces frontier-model context cost, it adds local and mid-level model calls, so practical efficiency depends on parallel serving. Empirically, the benefits are strongest for cumulative interaction memory; on long-document understanding, stronger RAG settings with larger K or chunk size can sometimes outperform HMARS. Finally, our diagnostic benchmarks are designed to stress evidence breadth and context-dependent relevance, and should be complemented with broader natural workloads and additional agentic-memory baselines in future work.

References

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

Akari Asai, Zeqiu Wu, Yizhong Wang, Avi Sil, and Hannaneh Hajishirzi. 2024. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *International conference on learning representations*, volume 2024, pages 9112–9141.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, and 1 others. 2024. Longbench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 3119–3137.

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, and 1 others. 2025. Longbench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3639–3664.

Haonan Bian, Zhiyuan Yao, Sen Hu, Zishan Xu, Shaolei Zhang, Yifu Guo, Ziliang Yang, Xueran Han, Huacan Wang, and Ronghao Chen. 2026. Realmem: Benchmarking llms in real-world memory-driven interaction. *arXiv preprint arXiv:2601.06966*.

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*.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

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*.

Shailja Gupta, Rajesh Ranjan, and Surya Narayan Singh. 2024. A comprehensive survey of retrieval-augmented generation (rag): Evolution, current landscape and future directions. *arXiv preprint arXiv:2410.12837*.

Yi Jiang, Sendong Zhao, Jianbo Li, Haochun Wang, Lizhe Zhang, Yan Liu, and Bing Qin. 2025. Cocoa: Collaborative chain-of-agents for parametric-retrieved knowledge synergy. *arXiv preprint arXiv:2508.01696*.

Ziyan Jiang, Xueguang Ma, and Wenhu Chen. 2024. Longrag: Enhancing retrieval-augmented generation with long-context llms. *arXiv preprint arXiv:2406.15319*.

Jiajie Jin, Xiaoxi Li, Guanting Dong, Yuyao Zhang, Yutao Zhu, Yongkang Wu, Zhonghua Li, Qi Ye, and Zhicheng Dou. 2025. Hierarchical document refinement for long-context retrieval-augmented generation. *Preprint*, arXiv:2505.10413.

Kuang-Huei Lee, Xinyun Chen, Hiroki Furuta, John Canny, and Ian Fischer. 2024. A human-inspired reading agent with gist memory of very long contexts. *arXiv preprint arXiv:2402.09727*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher Manning. 2024. Raptor: Recursive abstractive processing for tree-organized retrieval. In *International Conference on Learning Representations*, volume 2024, pages 32628–32649.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, Akshay

Nathan, Alan Luo, Alec Helyar, Aleksander Madry, Aleksandr Efremov, Aleksandra Spyra, Alex Baker-Whitcomb, Alex Beutel, Alex Karpenko, and 467 others. 2026a. Openai gpt-5 system card. *Preprint*, arXiv:2601.03267.

Aditi Singh, Abul Ehtesham, Saket Kumar, Tala Talaei Khoei, and Athanasios V. Vasilakos. 2026b. Agentic retrieval-augmented generation: A survey on agentic rag. *Preprint*, arXiv:2501.09136.

Haoran Sun, Shaoning Zeng, and Bob Zhang. 2026. H-mem: Hierarchical memory for high-efficiency long-term reasoning in llm agents. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 341–350.

Zhen Tan, Jun Yan, I-Hung Hsu, Rujun Han, Zifeng Wang, Long Le, Yiwen Song, Yanfei Chen, Hamid Palangi, George Lee, and 1 others. 2025. In prospect and retrospect: Reflective memory management for long-term personalized dialogue agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8416–8439.

Yu Wang and Xi Chen. 2025. Mirix: Multi-agent memory system for llm-based agents. *arXiv preprint arXiv:2507.07957*.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. 2026. A-mem: Agentic memory for llm agents. *Advances in Neural Information Processing Systems*, 38:17577–17604.

Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Jinhe Bi, Kristian Kersting, Jeff Z Pan, and 1 others. 2025. Memory-r1: Enhancing large language model agents to manage and utilize memories via reinforcement learning. *arXiv preprint arXiv:2508.19828*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, and 1 others. 2025. Memagent: Reshaping long-context llm with multi-conv rl-based memory agent. *arXiv preprint arXiv:2507.02259*.

Ningning Zhang, Xingxing Yang, Zhizhong Tan, Weiping Deng, and Wenyong Wang. 2026. Himem: Hierarchical long-term memory for llm long-horizon agents. *arXiv preprint arXiv:2601.06377*.

Yusen Zhang, Ruoxi Sun, Yanfei Chen, Tomas Pfister, Rui Zhang, and Sercan Ö Arık. 2024. Chain of agents: Large language models collaborating on long-context tasks. *Advances in Neural Information Processing Systems*, 37:132208–132237.

Jun Zhao, Can Zu, Xu Hao, Yi Lu, Wei He, Yiwen Ding, Tao Gui, Qi Zhang, and Xuan-Jing Huang. 2024. Longagent: achieving question answering for 128k-token-long documents through multi-agent collaboration. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 16310–16324.

Zijian Zhou, Ao Qu, Zhaoxuan Wu, Sunghwan Kim, Alok Prakash, Daniela Rus, Jinhua Zhao, Bryan Kian Hsiang Low, and Paul Pu Liang. 2025. Mem1: Learning to synergize memory and reasoning for efficient long-horizon agents. *arXiv preprint arXiv:2506.15841*.

A Benchmark Construction Details

A.1 Long-Document Benchmark Construction

This appendix provides additional details on the construction of the long-document understanding benchmark introduced in Section 4.1.1.

Source document selection. We derive the long-document benchmark from LongBench-v2, selecting 29 English documents between 20K and 120K tokens across legal texts, scientific papers, literary criticism, and game-theory transcripts.

Question generation. For each document, we prompt GPT-5.1 to generate candidate questions in the five categories defined in Section 4.1 (Synthesis, Comparison, Causal, Cross-reference, Inference). The generation prompt instructs the model to:

1. Identify 3–5 non-trivial facts spread across different sections.
2. Formulate a question whose correct answer requires integrating all identified facts.
3. Provide a gold-standard answer and list the minimal set of supporting passages (by section or paragraph index).

We generate 10–15 candidate questions per document to allow for attrition during quality control.

Supporting chunk annotation. We segment each document into 1,024-token chunks and map supporting passages to chunk indices. A question is retained only if it requires at least four supporting chunks.

Quality control and RAG recall filtering. We compute embedding-based RAG recall using the same embedding model as our RAG baseline. For each question, we retrieve K chunks, where K equals the number of annotated supporting chunks. We retain only questions with RAG recall ≤ 0.5. After filtering, the benchmark contains 195 questions.

Example.

- **Type:** Synthesis
- **Question:** "Based on Sections 3, 5, and 7 of the report, what are the three main factors contributing to the observed decline in biodiversity, and how do the authors propose to address each?"
- **Supporting chunks:** 12, 34, 58 (out of 72 total chunks)
- **Gold answer:** (3-sentence summary linking habitat loss, climate change, and invasive species to the proposed mitigation strategies.)

A.2 Cumulative Interaction Benchmark Construction

This appendix details the construction of the cumulative multi-turn dialogue benchmark introduced in Section 4.1.2.

Scenario design. We generate 120 dialogues across 30 realistic scenarios, including travel planning, apartment search, debugging, and task coordination.

Dialogue generation. Each dialogue contains 24–49 turns, with 5–9 key facts scattered throughout the interaction. Key facts include names, numbers, dates, preferences, constraints, and user updates.

Key fact placement. Key facts are embedded at pre-designated turns (e.g. turns 2, 5, 9, 12) so that the final probe question requires the model to recall and integrate information from multiple, temporally distant points in the conversation. On average, each dialogue contains 5.1 key facts spread across 4.2 distinct turns.

Subtype definitions. The 120 probe questions are evenly divided into three subtypes:

- **Fact Recall** (40 questions): Requires retrieving a specific fact stated in an earlier turn.
- **Temporal Ordering** (40 questions): Requires determining the chronological sequence of events or decisions discussed across turns.
- **Cross-Turn Inference** (40 questions): Requires combining facts from $\geq$2 non-adjacent turns to derive a conclusion not stated in any single turn.

Example.

- **Scenario:** Chronic disease management (Healthcare)
- **Turns:** 12 (average history length: 11.8 K tokens)
- **Probe (Cross-Turn Inference):** "Given the patient's updated lab results from turn 9 and the medication change discussed in turn 4, should the treatment plan be revised?"
- **Key facts:** Medication dosage (turn 4), lab results (turn 9), contraindication guideline (turn 2)

B Implementation Details and Hyperparameters

Model hierarchy. HMARS uses an asymmetric model hierarchy. Sub-agent local memory cards are generated with Qwen3-0.6B, mid-agent coordination and page selection are performed with Qwen3-8B, and final answer synthesis is performed by GPT-5.1 model. The Qwen models are served with vLLM. Unless otherwise stated, all model calls use temperature 0.1. We run all the experiments on NVIDIA H100 GPU.

Document partitioning. All splitting is performed with the model tokenizer rather than character length. Given a document or interaction history $\mathcal{D}$, we partition it into local memory regions of at most $C_{\text{sub}} = 16{,}384$ tokens. Each local region is assigned to one sub-agent and further divided into pages of size $P = 1024$ tokens. Each page stores its global token offsets, so selected pages can be mapped back to the original document and ordered before final synthesis.

Agent grouping. Sub-agents are grouped under mid-agents with maximum group size $G = 10$. If the document contains M sub-agents, the system creates $K = \lceil M/G \rceil$ mid-agents. The final group may contain fewer than G sub-agents. Each mid-agent constructs a shared regional memory from the local memory cards of its assigned sub-agents.

Query-time configuration. At query time, HMARS uses full-fanout page selection: every sub-agent receives the query and participates in local page assessment. Mid-agents do not serve as hard gates; instead, they produce a region-level query interpretation and sub-agent-specific hints. Sub-agents then select relevant page IDs conditioned on the query, their local memory card, and the mid-agent hint. Selected pages are deduplicated, sorted by global token offset, and passed to the frontier model as the final evidence context.

Baseline settings. For dense RAG, we use Qwen3-Embedding-0.6B with cosine similarity and the same frontier answer model as HMARS. We evaluate both 1K-token and 4K-token retrieval chunks with multiple retrieval budgets. For Dense Rerank, we first retrieve the top-N candidates using the same dense retriever, then score candidates with Qwen3-8B and pass the top-ranked chunks to the frontier model. Full-Context LLM directly feeds the document to the frontier model up to the model context limit. GraphRAG, Chain-of-Agents, and MemAgent follow their standard or released inference protocols when applicable.

Evaluation. We use GPT-based judging following LongBench-v2. For each question, gold answer, and predicted answer, the judge assigns a binary correctness score with a short justification. Accuracy is the primary metric. For efficiency analysis, we report latency, frontier prompt tokens, and estimated per-query FLOPs. Prompt tokens refer to the text passed to the frontier answer model, not the full text processed by lightweight sub-agents or mid-agents.

C Why Full-Context Consideration Helps

Consider a query q whose answer requires evidence from a set of supporting pages or chunks $\mathcal{E} = \{e_1, \ldots, e_R\}$ distributed across the input. In a standard RAG pipeline, each chunk c_i is scored independently by a similarity function $\text{sim}(q, c_i)$, and only the top-K chunks are passed to the answer model. The evidence recall is

$$\text{Recall}_{\text{RAG}} = \frac{|\{e \in \mathcal{E} : e \in \text{top-}K\}|}{|\mathcal{E}|}. \quad (8)$$

This mechanism can fail even when the evidence is present in the input. A passage may use vocabulary different from the query, or its relevance may depend on information stated elsewhere. For example, a question about "housing constraints" may require an earlier turn mentioning a "$2,500 monthly budget," despite limited lexical overlap.

HMARS removes this early exclusion bottleneck by giving every local memory region an opportunity to perform query-conditioned assessment. This should not be interpreted as guaranteeing perfect evidence recall: local agents may still fail to select a relevant page. Rather, the system guarantees *consideration coverage*: no region is discarded before a local agent evaluates its pages under the query and the mid-agent's regional guidance.

This design is most useful in two settings. First, in *evidence-breadth* questions, the answer depends on many non-adjacent regions, so a small top-K set is unlikely to contain all supporting evidence. Second, in *context-dependent relevance* questions, whether a passage matters can only be determined after the query is interpreted against broader memory state. These are precisely the settings targeted by our diagnostic benchmarks.

D Computational Complexity

Let N denote the document length in tokens, P the page size, C_{sub} the maximum token capacity of each sub-agent, and G the maximum number of sub-agents assigned to one mid-agent. The number of sub-agents and mid-agents are

$$M = \left\lceil \frac{N}{C_{\text{sub}}} \right\rceil, \qquad K = \left\lceil \frac{M}{G} \right\rceil. \tag{9}$$

Indexing cost. During indexing, each sub-agent summarizes its assigned region into a local memory card, and each mid-agent aggregates a group of local memory cards into shared regional memory. If C_{sum} and C_{reg} denote the average token costs of the local and regional summarization calls, then the indexing cost scales as

$$\text{FLOPs}_{\text{index}} \approx 2|\theta_{\text{sub}}|MC_{\text{sum}} + 2|\theta_{\text{mid}}|KC_{\text{reg}}. \tag{10}$$

This cost is amortized across all queries over the same document or interaction history.

Query cost. At query time, each mid-agent produces a regional query interpretation and sub-agent-specific hints. Then local page selection is performed for each sub-agent, followed by one frontier-model synthesis call. Let θ_{sel} denote the model used for page selection; in our implementation, this is the Qwen3-8B model used for mid-level coordination. The query cost scales as

$$\begin{aligned}\text{FLOPs}_{\text{query}} \approx\ & 2|\theta_{\text{mid}}|KC_{\text{broadcast}} \\ & + 2|\theta_{\text{sel}}|MC_{\text{select}} \\ & + 2|\theta_{\text{base}}|C_{\text{ans}}.\end{aligned} \tag{11}$$

The main efficiency advantage comes from asymmetric model allocation. Full-context consideration is performed by smaller local and mid-level models, while the frontier model only receives the selected grounded evidence.

Parallelism. The mid-agent calls and sub-agent selection calls can be executed concurrently within each phase. In practice, latency is dominated by the slowest local or mid-level call plus the final frontier-model synthesis call, rather than by the sum of all local calls.

E Additional Baseline Notes

Dense retrieval with reranking. Dense Rerank is the strongest flat retrieval baseline in our experiments. It first retrieves a large candidate set using the same embedding model as RAG, then scores candidates with Qwen3-8B before passing the top-ranked chunks to the frontier answer model. Its strong performance supports one of our main design intuitions: small-model relevance scoring is a useful primitive. However, it remains a flat candidate-ranking pipeline. It does not assign persistent local memory ownership, does not construct shared regional memory, and does not allow all memory regions to participate in query-conditioned assessment.

RAG chunk size and retrieval budget. We evaluate RAG with both 1K and 4K retrieval chunks. These chunk sizes play a different role from HMARS page size. A RAG chunk is both the retrieval unit and the text passed to the answer model, so larger chunks may preserve more local context but also dilute retrieval precision and increase prompt noise. In contrast, a HMARS page is a grounded evidence unit selected within a larger agent-managed memory region. This distinction explains why increasing RAG's chunk size does not consistently improve performance across our benchmarks.

Full-context prompting. Full-context prompting gives the frontier model access to more tokens, but it does not provide an explicit memory interface. The model must still locate scattered evidence among distractors. Our results show that exposing the full document to the frontier model is not a reliable upper bound for long-context reasoning.

GraphRAG. GraphRAG provides an entity-centric abstraction over documents and performs local search over an extracted graph. This can be useful when questions are primarily entity-centric, but graph/community summaries may lose fine-grained passage-level evidence needed for broad cross-region synthesis.

Agentic long-context baselines. Chain-of-Agents and MemAgent represent sequential agentic processing and rolling-memory approaches. They demonstrate that agentic reading can extend beyond a single prompt, but they compress information into communication units or evolving memory states. In our diagnostic benchmarks, this compression can discard fine-grained evidence. HMARS instead preserves grounded page access throughout the hierarchy.

Why CoA and MemAgent underperform in our setting. CoA-7B and MemAgent-14B represent sequential agentic long-context processing, but their design assumptions differ from the diagnostic setting studied here. Both methods compress information during a single pass over the input: CoA propagates a communication unit across chunk-level workers, while MemAgent maintains an evolving memory state. This compression is useful for reducing context length, but can lose fine-grained evidence when answers require multiple non-adjacent supporting passages or when an early passage becomes relevant only after later context is observed. In representative errors, the required evidence was often present in the input but absent from the information available to the final answer model. Thus, these results should not be read as showing that CoA or MemAgent are weak methods in general; rather, they illustrate a mismatch between sequential compression-based reading and our evidence-breadth, context-dependent memory setting. HMARS is designed for this failure mode by preserving grounded page access and allowing all local memory regions to participate in query-conditioned assessment.

F Prompt Details

Prompt 3: AnswerPrompt

You are an expert reader. Answer the question STRICTLY based on the retrieved passages below. If the answer is not supported by the passages, say so explicitly. Be concise (1–4 sentences) and factual.

Retrieved passages:

`{context}`

Question:

`{question}`

Answer:

Prompt 4: LocalMemoryCardPrompt

You are a local memory agent for one section of a long document or interaction history.

Below are the pages in your assigned section. Write a concise memory card that captures the main topics, entities, events, constraints, numbers, and distinctive facts in this section. Be specific about what this section contains, since a mid-level agent will use your memory card to coordinate future queries.

Pages:

`{pages_text}`

Return ONLY a JSON object with the following fields:

`{JSON_form}`

Prompt 5: BroadcastPrompt

You are a mid-level memory coordinator. You manage several local memory regions.

Your regional summary:

`{region_summary}`

Your local regions, with topics and entities:

`{sub_map_text}`

Query:

`{query}`

Interpret how the query may relate to this region. Then select every local region whose content could plausibly contain useful evidence. Do NOT artificially limit the number of selected regions. For each selected region, write a focused hint telling the local agent what to look for, including relevant entities, numbers, constraints, temporal cues, or sub-topics.

Return ONLY a JSON object of this form:

`{JSON_form}`

Prompt 6: SelectPrompt

You are a local memory agent for one section of a long document or interaction history. Select every page from your section that could plausibly help answer the query. Be greedy: when in doubt, include the page. Do not limit the number of pages you return.

Regional context from your mid-level coordinator:

`{region_summary}`

What the coordinator believes your section contains:

topics: `{sub_topics}`

entities: `{sub_entities}`

Focused hint from your coordinator:

`{hint}`

Query:

`{query}`

Pages in your section:

`{pages_text}`

Return ONLY a JSON object of this form:

`{JSON_form}`

G Dataset and Artifact Documentation

Data sources. Our long-document benchmark is derived from LongBench-v2, and our cumulative interaction benchmark is inspired by RealMem. We use these sources only for research evaluation and

follow their intended benchmark usage.

Dataset construction. We describe the document selection, question generation, filtering criteria, supporting-evidence annotation, dialogue scenario generation, and task subtype definitions in Appendix A.

Statistics. The final benchmark contains 195 long-document questions and 120 multi-turn interaction questions. The long-document benchmark contains 29 documents with lengths between 20K and 120K tokens, and each retained question requires at least four supporting chunks. The multi-turn benchmark contains 120 dialogues across 30 scenarios, with 24–49 turns per dialogue and 5–9 key facts per sample.

PII and offensive content. The constructed evaluation data consists of benchmark documents and synthetic task-oriented dialogues. We do not intentionally include personally identifying information or offensive content. For synthetic dialogues, names, dates, constraints, and preferences are fictional.

H Licenses and Intended Use

Licenses. Our benchmark construction uses LongBench-v2 and RealMem-style settings as research artifacts. We cite the original creators in the main paper and use the data only for academic evaluation. Users should follow the licenses and usage terms of the original benchmark sources.

Intended use. The constructed evaluation sets are intended for evaluating long-context question answering, evidence-breadth reasoning, and multi-turn interaction memory. They are not intended for training deployed assistants, profiling real users, or making decisions about individuals.

Release format. We will release the evaluation data in JSONL format, including the input document or dialogue, question, gold answer, supporting evidence metadata, and task subtype when applicable.

I Use of AI Assistants

We also used LLMs to generate candidate questions and synthetic dialogues during benchmark construction. All generated examples were filtered according to the criteria described in Appendix A.

J Human Subjects

This work does not involve human subjects, recruited participants, or paid annotators. The multi-turn interaction benchmark consists of synthetic task-oriented dialogues generated for evaluation purposes. Therefore, participant instructions, recruitment, payment, consent, and IRB approval are not applicable.