COSMIR: Chain Orchestrated Structured Memory for Iterative Reasoning over Long Context

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

Reasoning over very long inputs remains difficult for large language models (LLMs). Common workarounds either shrink the input via retrieval (risking missed evidence), enlarge the context window (straining selectivity), or stage multiple agents to read in pieces. In staged pipelines (e.g., Chain of Agents, CoA), free-form summaries passed between agents can discard crucial details and amplify early mistakes. We introduce COSMIR (**C**hain **O**rchestrated **S**tructured **M**emory for **I**terative **R**easoning), a chain-style framework that replaces ad hoc messages with a structured memory. A PLANNER agent first turns a user query into concrete, checkable sub-questions. WORKER agents process chunks via a fixed micro-cycle: Extract, Infer, Refine, writing all updates to the shared memory. A MANAGER agent then SYNTHESIZES the final answer directly from the memory. This preserves stepwise read-then-reason benefits while changing both the communication medium (structured memory) and the worker procedure (fixed micro-cycle), yielding higher faithfulness, better long-range aggregation, and auditability. On long-context QA from the HELMET suite, COSMIR reduces propagation-stage information loss and improves accuracy over a CoA baseline.

1 Introduction

Large Language Models (LLMs) have rapidly advanced language understanding and generation tasks, supporting assistants, search, and retrieval systems [Wu et al., 2024, LangChain, 2023]. However, reasoning over *long* inputs, for example, books, extended technical documents, or large code repositories, remains brittle [Liu et al., 2024, Brown et al., 2020, Srivastava et al., 2023]. Mitigation strategies typically follow two paths. The first contracts the input through retrieval (e.g., RAG [Lewis et al., 2020]), which can omit crucial evidence and inject noise. The second expands the model context windows [Peng et al., 2024], but still struggles with selectivity [Liu et al., 2024] and faces practical scaling limits [Wang et al., 2024a].

A complementary line decomposes long-context reasoning into steps executed over shorter spans. This includes tree- or graph-structured prompting [Yao et al., 2023] and multi-agent coordination [Zhang et al., 2024b]. Although effective, sequential pipelines that pass *free-form summaries* between steps are vulnerable to compression loss and cascading errors. An agent must spot what matters in its local fragment, compress it into an ad hoc message, and anticipate future relevance. Omissions or imprecisions early on can silently propagate and degrade the final answer (Appendix A.1).

We propose COSMIR, ***C**hain **O**rchestrated **S**tructured **M**emory for **I**terative **R**easoning*, a training-free framework that keeps the stepwise "read–reason" benefit while replacing free text messages with a *structured, centralized working memory*. A PLANNER converts the user query into concrete, checkable sub-questions. WORKERS traverse chunks using a fixed micro-cycle: EXTRACT evidence under a memory budget, INFER grounded claims from accumulated evidence, and REFINE the unresolved

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

question set. The worker then writes the information into a shared memory M. A Manager then Synthesizes the final answer directly from M. This design reduces propagation stage information loss, improves long-range aggregation, and yields an auditable trace of how the answer was produced.

Our key contributions are: 1] We introduce COSMIR, a training-free, interpretable framework for long-context reasoning that replaces free-form message passing with a centralized memory and a fixed worker micro-cycle. 2] We show in long-context QA benchmarks (HELMET suite [Yen et al., 2025]) that COSMIR reduces information loss and improves accuracy over a Chain of Agents (CoA) [Zhang et al., 2024b] baseline at comparable cost.

Paper organization. Section 2 analyzes a representative failure of CoA [Zhang et al., 2024b] due to propagation stage information loss and illustrates how COSMIR prevents it. Section 3 situates our work among long-context modeling, multi-agent prompting, and structured memory. Section 4 formalizes COSMIR end-to-end describing both the structured, centralized memory and the different agent executions. Sections 5 and 6 describe the experimental setup and results. We discuss limitations and future work before concluding.

2 Example Case Study

Figure 1 illustrates COSMIR on a question from the **Infbench-QA** dataset (Section 5): *Where did Kiara and Carter first meet before becoming roommates in Nigeria?* Early in the book (chunk 1), the text states that **Kiara met a pale young gentleman at Miss Kiley's house; they fought in the garden**, without naming the gentleman. Much later (chunk R), the gentleman is identified as Carter.

With COSMIR, the Planner seeds "Questions" with targeted set of sub-questions such as **What is Kiara's history of encounters before becoming roommates with Carter?**. The Extract phase of Worker records the early passage as a relevant element in "Gathered Facts" (preserving the text under the memory budget). When a later fragment reveals the identity of the gentleman, the Infer phase of the Worker reconciles the two sections into an entry in "Inferred Facts", resolving the ambiguity of the cross reference. Refine phase marks the relevant sub-question as answered and prunes distractors. Finally, Manager composes the answer using both the early encounter span and the later identity span in "Structured Memory", resulting in a faithful and evidence-cited resolution.

By contrast, pipelines that rely on unstructured summaries (e.g., CoA-style message passing) frequently compress away the unnamed encounter or fail to reconnect it when the identity appears many chunks later, leading to missed long-range links. The failure example is provided in more detail in Appendix A.1.1.

3 Related Work

We review three areas relevant to our framework: long-context modeling, multi-agent collaboration, and memory mechanisms.

Long-Context Modeling for LLMs Extending the context window remains a core challenge. Techniques like Retrieval-Augmented Generation (RAG) [Lewis et al., 2020] aim to reduce the input by retrieving relevant segments via embedding similarity but often miss critical evidence [Zhang et al., 2024b]. Window-extension methods aim to extend LLM context windows using new attention mechanisms [Liu and Abbeel, 2023] and position interpolation [Peng et al., 2024]. Such methods, along with large-context models such as Claude Sonnet 4 [Anthropic, 2024], enable direct processing but suffer from degraded focus [Liu et al., 2024]. Recent proposals like MemAgent [Yu et al., 2025] and Sculptor [Li et al., 2025] explore memory-augmented processing, but do not explicitly structure reasoning dependencies. Parallel works in improving model reasoning capabilities over long-context, like SELF-DISCOVER [Zhou et al., 2024], ALR^2 [Li et al., 2024], and InfinityThink [Yan et al., 2025], enhance reasoning by adopting explicit task-specific structure and decouple reasoning and inferencing. However, these approaches rely on the base model to jointly perform reasoning, inference, and memory management, which can overextend its capacity in long-context scenarios. COSMIR adopts benefits from structured reasoning and augments them with memory-augmented processing by enforcing explicit state-based structured reasoning.

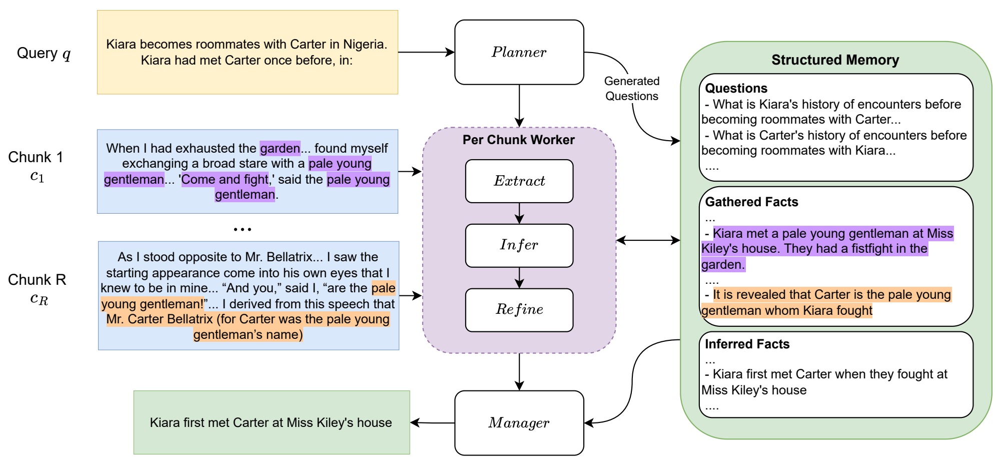

Figure 1: Overview of COSMIR, a training-free framework for long context tasks. It consists of a Planner agent which given the question generates clarifying sub-questions. Segmented chunks from the context are then processed by the Worker agent in a fixed micro cycle which has three phases: Extract, Infer and Refine. Through these phases the Worker agent edits a structured centralized memory by extracting facts, making logical inferences over the facts and planning next steps by removing/adding new sub-questions. Finally, the structured memory is passed to the Manager agent to generate a final coherent answer. Boxes in blue are excerpts from chunks c_1 and c_R. Key portions of these excerpts, that are needed to answer the query q have been highlighted and corresponding facts that have been extracted from these chunks have been highlighted in the structured memory.

Multi-Agent LLM Collaboration Multi-agent systems have been widely studied for decomposing complex tasks [Guo et al., 2024]. Prior work in the space of multi-agent LLM collaboration focuses on reasoning on small text through multi-agent discussion [Du et al., 2024, Xiong et al., 2023, Chen et al., 2023b, Tang et al., 2024, Chen et al., 2024, Zhao et al., 2024] on domains like reasoning [Du et al., 2024, Tang et al., 2024, Zhao et al., 2024], paper review [Xu et al., 2023], and coding generation [Wang et al., 2025, Wadhwa et al., 2024]. Different from prior works, we target reasoning over long contexts. For long context reasoning, XpandA [Xiao et al., 2025] uses dynamic chunk partitioning and selective replay mechanisms to accelerate inference on long texts. Other collaborative strategies (e.g., multi-hop prompting [Yao et al., 2023]) improve decomposition but lack persistent, structured memory. COSMIR differs by combining multi-agent decomposition with centralized, structured memory. Among prior multi-agent approaches for long-context handling, our work is most similar to Chain-of-Agents (CoA) [Zhang et al., 2024b], which handles long-context reasoning by coordinating multiple worker agents in sequential collaboration but this can result in information loss and cascading errors. COSMIR improves on this approach by conditioning the workers with high quality clarifying questions and decomposing the worker agent into multiple phases which focus on dedicated subtasks, namely; Fact Extraction, Logical Inference and Problem Refinement.

Memory Systems for LLMs Memory systems for LLMs have been explored in several dimensions. Training time approaches integrate memory directly into the architecture, such as recurrent memory layers [Bulatov et al., 2022], side network memory encoders [Wang et al., 2023, 2024b], or through trainable memory layers [Berges et al., 2025]. Other training methods involve training models to generate designated memory tokens [Jin et al., 2025, Yu et al., 2025, Qian et al., 2025]. Runtime methods instead attach external stores [Zhong et al., 2024, Das et al., 2024] or retrieve memory units created from the token sequence [Xiao et al., 2024, Fountas et al., 2025]. More structured approaches explicitly organize and manage memory contents: for instance, MemWalker [Chen et al., 2023a] generates, organizes, and consumes hierarchical summaries of the context, while HippoRAG [Gutiérrez et al., 2025] takes inspiration from associative memory in the brain, building graph-like structures that support spreading activation and relational retrieval. While these systems enhance retention and retrieval, they often operate at a hidden or architectural level, limiting interpretability.

Our method complements them by providing a transparent, text-based memory that explicitly records gathered, inferred facts and unanswered threads, enabling workers to reason collaboratively while exposing intermediate states for inspection.

4 Methodology

COSMIR inherits the high-level idea from CoA [Zhang et al., 2024b] of traversing long contexts with lightweight agents, but generalizes it in two ways: (1) a planner that converts the user query into concrete investigation targets and (2) workers who operate over a structured centralized working memory rather than emitting free-form summaries. The manager then produces the final answer from that structured memory. This preserves the intuition of chain-style processing while changing both the artifact being passed (structured memory vs. summary) and the internal procedure of the worker (a fixed micro-cycle instead of "summarize and pass").

4.1 Centralized Memory

The centralized memory in COSMIR is defined as

$$M := \left\langle \mathcal{Q}, \mathcal{F}_g, \mathcal{F}_i, a \right\rangle, \tag{1}$$

where $\mathcal{Q}$ denotes the set of unresolved sub-questions, $\mathcal{F}_g$ the set of gathered facts, $\mathcal{F}_i$ the set of inferred facts, and a the synthesized answer, which remains empty until the reasoning process terminates. To limit the context available to each agent, the size of $\mathcal{F}_g$ is constrained to at most a k-fraction of the length of a chunk.

4.2 Agent Roles and Execution

PLANNER (Decompose). From the user query q, the planner seeds $\mathcal{Q}$ with a small set of checkable questions. The PLANNER generates two classes of questions; Focused questions that decompose the user query q into smaller bite-sized sub-questions and exploratory information nets that promote broad fact extraction which serves to catch facts that might slip through the direct questions.

WORKER (Analyze chunks with a fixed micro-cycle). Given a chunk c_j and current M, each worker performs a three-step micro-cycle:

- EXTRACT: From chunk c_j and the current question set $\mathcal{Q}$, select evidence units relevant to the user query q and the current question set $\mathcal{Q}$ and append them to $\mathcal{F}_g$ while adhering to the memory budget. If the size of $\mathcal{F}_g$ goes over the allotted memory budget then oldest facts in $\mathcal{F}_g$ are pruned away till $\mathcal{F}_g$ fits in the allotted budget.
- INFER: Using $E = \mathcal{F}_g \cup \mathcal{F}_i$, derive new, grounded claims and add them to $\mathcal{F}_i$.
- REFINE: Update the question set $\mathcal{Q}$ by marking resolved items and spawning focused follow-ups that improve utility of later chunks.

MANAGER (Synthesize). After all chunks are processed, the manager computes a using the memory M producing the answer plus an optional rationale citing evidence.

Algorithm 1 provides the end-to-end pseudo-code for COSMIR, detailing planning, the worker micro-cycle (EXTRACT, INFER, REFINE) over chunks, and the final synthesis by the manager.

5 Experimental Setup

5.1 Datasets

We evaluate COSMIR on the long context QA split of the HELMET benchmark [Yen et al., 2025]. This split consists of three datasets, namely:

1. $\infty bench$ English QA: This dataset consists of freeform questions on English novels with entity replacement. The evaluation metric is ROUGE F1 score [Lin, 2004]. We refer to this dataset as **InfBench-QA** going forward.

Algorithm 1 COSMIR: **C**hain **O**rchestrated **S**tructured **M**emory for **I**terative **R**easoning

Require: query q; chunks $C = \{c_1, \dots, c_L\}$; memory fraction k
Ensure: answer a
$\mathcal{Q} \leftarrow \textsc{Plan}(q)$; ▷ Planner agent
$\mathcal{F}_g \leftarrow \varnothing$; $\mathcal{F}_i \leftarrow \varnothing$; $a \leftarrow \varnothing$
$M \leftarrow \langle \mathcal{Q}, \mathcal{F}_g, \mathcal{F}_i, a \rangle$
for $j = 1$ to L **do** ▷ Worker agents process chunks left-to-right
  $\Delta\mathcal{F}_g \leftarrow \textsc{Extract}(c_j, \mathcal{Q})$
  $\mathcal{F}_g \leftarrow \mathcal{F}_g \cup \Delta\mathcal{F}_g$
  $\mathcal{F}_g \leftarrow \textsc{Prune}(\mathcal{F}_g, k)$
  $\Delta\mathcal{F}_i \leftarrow \textsc{Infer}(\mathcal{F}_g, \mathcal{F}_i)$
  $\mathcal{F}_i \leftarrow \mathcal{F}_i \cup \Delta\mathcal{F}_i$
  $\mathcal{Q} \leftarrow \textsc{Refine}(\mathcal{Q}, \mathcal{F}_g, \mathcal{F}_i)$
  $M \leftarrow \langle \mathcal{Q}, \mathcal{F}_g, \mathcal{F}_i, a \rangle$ ▷ Structured communication unit
end for
$a \leftarrow \text{SYNTHESIZE}(M)$ ▷ Manager agent
return a

2. $\infty bench$ English MC: This dataset consists of multiple-choice questions on English novels with entity replacement. The evaluation metric is exact match (EM). We refer to this dataset as **InfBench-MC** going forward.
3. NarrativeQA: This dataset consists of free-form questions on English books and movie scripts. The evaluation metric is ROUGE F1 score [Lin, 2004].

Specifically, for NarrativeQA, we further filter the dataset to only have questions with a context of at least 256000 tokens; we call this subset **NarrativeQA-256k**.

5.2 Baselines and System Configurations

The primary baseline that we test COSMIR against is CoA [Zhang et al., 2024b]. For both COSMIR and CoA, a chunk size of 64000 tokens is chosen, while the maximum size of the summary and memory is chosen to be 8000 tokens. We additionally also test COSMIR against a truncated context setting (TC) where the context is truncated down to 128000 tokens by removing sentences from the middle of the context [Zhang et al., 2024a].

We run all three techniques with three models: *GPT-4.1*, *GPT-4.1-mini*, and *Qwen3-14B*. Model-level settings (temperature, max tokens) are identical across methods to ensure fair comparison.

6 Results and Analysis

Table 1 shows the results for the three models for all three datasets. We see that COSMIR outperforms both baselines for all model-dataset combinations. The largest gains of COSMIR over CoA are seen for **InfBench-QA** and **NarrativeQA-256k**, which are free-form question-response benchmarks. The gains are also consistent across different model sizes, showing the universal applicability of the technique.

Performance gains of COSMIR and CoA over the TC baseline are representative of the better extraction and storage of facts in both CoA and COSMIR. Furthermore, the TC baseline illustrates performance degradation of models at extreme context lengths. This effect is especially pronounced for GPT-4.1-mini, which sees a steeper decline in performance compared to other models, consistently performing worse than both GPT-4.1 and Qwen3-14B for all the datasets in the TC baseline.

Gains between COSMIR and CoA are primarily driven by the decomposition of the reasoning process and the specific structured memory of COSMIR. The structured memory preserves far more contextual information than intermediate CoA summaries, resulting in lower information loss. Furthermore, generating targeted sub-questions helps guide the fact-extraction process, enabling the extraction of broader facts from the initial chunks. These facts can then serve both as input and contextual support for fact extraction and inference in later chunks. Both COSMIR and CoA have

Model	Method	InfBench-QA (ROUGE-F1)	InfBench-MC (Exact Match)	NarrativeQA-256k (ROUGE-F1)
GPT-4.1	TC	36.05	70.31	28.87
	CoA	47.62	86.03	35.27
	COSMIR	**50.74**	**87.33**	**37.58**
GPT-4.1-mini	TC	17.59	46.28	18.10
	CoA	40.47	72.49	29.17
	COSMIR	**43.56**	**74.23**	**31.43**
Qwen3-14B	TC	35.99	56.33	27.37
	CoA	38.12	65.07	29.53
	COSMIR	**40.76**	**65.93**	**31.14**

Table 1: Performance comparison of COSMIR, CoA, and TC across three long-context datasets for *GPT-4.1*, *GPT-4.1-mini*, and *Qwen3-14B*. The evaluation metrics for each dataset are mentioned alongside the dataset. Best results for each dataset and model are in **bold**.

high performance on the **InfBench-MC** benchmark. The multiple-choice options present with the query provide enough context for both techniques to correctly gather relevant evidence from the text. This also explains the meager gains seen between COSMIR and CoA.

As with sequential processing methods like CoA, fact extraction is the most critical component of COSMIR. If a relevant fact is not correctly extracted, later workers have no reliable way to reconstruct it unless the fact reappears elsewhere in the text. The remaining components in COSMIR are explicitly intended to support fact extraction. They produce high-quality clarifying questions to condition the Extract phase of the Worker and separate logical fact inference and problem-refinement into dedicated phases, but the Extract phase of the Worker remains the key bottleneck in the performance of COSMIR. We confirm this point with a targeted ablation, we initialize the Planner, the Infer phase of the Worker, the Refine phase of the Worker, and Manager agents with *GPT-4.1* while the Extract phase of the worker uses *Qwen3-14B* (we call this COSMIR-Extract-Qwen3) and compare the end task performance with initializing all components with *GPT-4.1* (COSMIR-GPT-4.1) and *Qwen3-14B* (COSMIR-Qwen3). Table 2 shows the results of these three configurations on the HELMET Long-Context QA benchmarks. We find that COSMIR-Extract-Qwen3 ablation performs better than COSMIR-Qwen3, especially for the **InfBench-QA** and **NarrativeQA-256k** benchmarks, but it falls quite short of the performance of COSMIR-GPT-4.1. The gains over COSMIR-Qwen3 are primarily driven by the higher quality of the other components in COSMIR-Extract-Qwen3. Just by reducing the quality of the Extract phase of the Worker in COSMIR-GPT-4.1, the performance has regressed closer to the performance of COSMIR-Qwen3, showing that the performance is bottlenecked by the quality of the Extract phase of the Worker agent.

Method	InfBench-QA (ROUGE-F1)	InfBench-MC (Exact Match)	NarrativeQA-256k (ROUGE-F1)
COSMIR-Qwen3	40.76	65.93	31.14
COSMIR-Extract-Qwen3	42.81	65.50	32.37
COSMIR-GPT-4.1	**50.74**	**87.33**	**37.58**

Table 2: Results for the HELMET long-context QA split for different model configurations. COSMIR-Qwen3 has all agents use *Qwen3-14B*, COSMIR-Extract-Qwen3 has the Extract phase of the Worker agent use Qwen3-14B while all other components use *GPT-4.1* and COSMIR-GPT-4.1 has all sub-agents use *GPT-4.1*

7 Limitations and Future Work

COSMIR improves evidence aggregation over CoA for long-context reasoning by combining specialized sub-agents with structured memory. However, the method depends critically on extraction quality: missed or low-quality extractions are difficult for later agents to recover and can limit end-task performance. COSMIR also increases per-example orchestration and requires thrice as many LLM calls as CoA. Future work can explore strategies to reduce the overall cost, for example, mixing models of different per-token costs to handle different parts of the COSMIR pipeline. Another limitation of the current experiments is that they rely on fixed-length chunks processed in their original order. Further analysis could investigate dynamic chunking strategies and approaches for determining optimal chunks and an effective ordering of those chunks, potentially revealing ways to improve performance even further. Finally, the current evaluation focuses on Long-Context QA benchmarks, the behaviour of COSMIR on other tasks and domains (e.g., summarization, legal/medical text) requires additional study. Extending the technique to a broader set of domains and addressing the extraction bottleneck more efficiently are promising directions for future work.

8 Conclusion

We presented COSMIR, a multi-stage agent architecture that decomposes long-context reasoning into explicit sub-tasks (Planning, Extract, Infer, Refine, Manager) and accumulates evidence in a structured memory separating gathered and inferred facts. In our evaluations, COSMIR improves long-context QA performance relative to chain-of-agents and truncated-context baselines while providing interpretable intermediate artifacts that reveal how evidence was collected and combined.

References

Anthropic. Claude sonnet 4, 2024. URL https://www.anthropic.com/claude/sonnet.

Vincent-Pierre Berges, Barlas Oguz, Daniel HAZIZA, Wen tau Yih, Luke Zettlemoyer, and Gargi Ghosh. Memory layers at scale. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=ATqGm1WyDj.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Aydar Bulatov, Yuri Kuratov, and Mikhail Burtsev. Recurrent memory transformer. In *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=Uynr3iPhksa.

Howard Chen, Ramakanth Pasunuru, Jason Weston, and Asli Celikyilmaz. Walking down the memory maze: Beyond context limit through interactive reading, 2023a. URL https://arxiv.org/abs/2310.05029.

Huaben Chen, Wenkang Ji, Lufeng Xu, and Shiyu Zhao. Multi-agent consensus seeking via large language models, 2023b. URL https://arxiv.org/abs/2310.20151.

Justin Chen, Swarnadeep Saha, and Mohit Bansal. ReConcile: Round-table conference improves reasoning via consensus among diverse LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024. URL https://aclanthology.org/2024.acl-long.381/.

Payel Das, Subhajit Chaudhury, Elliot Nelson, Igor Melnyk, Sarathkrishna Swaminathan, Sihui Dai, Aurelie Lozano, Georgios Kollias, Vijil Chenthamarakshan, Jiri Navratil, Soham Dan, and Pin-Yu

Chen. Larimar: Large language models with episodic memory control. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=t8mt4YrPsq.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the 41st International Conference on Machine Learning*, 2024. URL https://dl.acm.org/doi/10.5555/3692070.3692537.

Zafeirios Fountas, Martin Benfeghoul, Adnan Oomerjee, Fenia Christopoulou, Gerasimos Lampouras, Haitham Bou Ammar, and Jun Wang. Human-inspired episodic memory for infinite context LLMs. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=BI2int5SAC.

Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. Large language model based multi-agents: a survey of progress and challenges. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, 2024. URL https://www.ijcai.org/proceedings/2024/890.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From rag to memory: Non-parametric continual learning for large language models, 2025. URL https://arxiv.org/abs/2502.14802.

Mingyu Jin, Weidi Luo, Sitao Cheng, Xinyi Wang, Wenyue Hua, Ruixiang Tang, William Yang Wang, and Yongfeng Zhang. Disentangling memory and reasoning ability in large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025. URL https://aclanthology.org/2025.acl-long.84/.

LangChain. LangChain: Building applications with llms through composability, 2023. URL https://www.langchain.com/.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, 2020. URL https://dl.acm.org/doi/abs/10.5555/3495724.3496517.

Huayang Li, Pat Verga, Priyanka Sen, Bowen Yang, Vijay Viswanathan, Patrick Lewis, Taro Watanabe, and Yixuan Su. Alr^2: A retrieve-then-reason framework for long-context question answering, 2024. URL https://arxiv.org/abs/2410.03227.

Mo Li, L. H. Xu, Qitai Tan, Ting Cao, and Yunxin Liu. Sculptor: Empowering llms with cognitive agency via active context management, 2025. URL https://arxiv.org/abs/2508.04664.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, 2004. URL https://aclanthology.org/W04-1013/.

Hao Liu and Pieter Abbeel. Blockwise parallel transformers for large context models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1bfd87d2d92f0556819467dc08034f76-Abstract-Conference.html.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 2024. URL https://aclanthology.org/2024.tacl-1.9/.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. YaRN: Efficient context window extension of large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=wHBfxhZu1u.

Hongjin Qian, Zheng Liu, Peitian Zhang, Kelong Mao, Defu Lian, Zhicheng Dou, and Tiejun Huang. Memory never fades: Boosting long context processing with global memory-enhanced retrieval augmentation. In *The Web Conference 2025*, 2025. URL https://openreview.net/forum?id=8Cggwrvkho.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adri Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*, 2023. URL https://openreview.net/forum?id=uyTL5Bvosj.

Xiangru Tang, Anni Zou, Zhuosheng Zhang, Yilun Zhao, Xingyao Zhang, Arman Cohan, and Mark Gerstein. MedAgents: Large language models as collaborators for zero-shot medical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024. URL https://aclanthology.org/2024.findings-acl.33/.

Nalin Wadhwa, Atharv Sonwane, Daman Arora, Abhav Mehrotra, Saiteja Utpala, Ramakrishna B Bairi, Aditya Kanade, and Nagarajan Natarajan. MASAI: Modular architecture for software-engineering AI agents. In *NeurIPS 2024 Workshop on Open-World Agents*, 2024. URL https://openreview.net/forum?id=NSINt8lLYB.

Bing Wang, Changyu Ren, Jian Yang, Xinnian Liang, Jiaqi Bai, LinZheng Chai, Zhao Yan, Qian-Wen Zhang, Di Yin, Xing Sun, and Zhoujun Li. MAC-SQL: A multi-agent collaborative framework for text-to-SQL. In *Proceedings of the 31st International Conference on Computational Linguistics*, 2025. URL https://aclanthology.org/2025.coling-main.36/.

Weizhi Wang, Li Dong, Hao Cheng, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. Augmenting language models with long-term memory. In *Advances in Neural Information Processing Systems*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/ebd82705f44793b6f9ade5a669d0f0bf-Abstract-Conference.html.

Xindi Wang, Mahsa Salmani, Parsa Omidi, Xiangyu Ren, Mehdi Rezagholizadeh, and Armaghan Eshaghi. Beyond the limits: A survey of techniques to extend the context length in large language models. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, 2024a. URL https://www.ijcai.org/proceedings/2024/0917.

Yu Wang, Yifan Gao, Xiusi Chen, Haoming Jiang, Shiyang Li, Jingfeng Yang, Qingyu Yin, Zheng Li, Xian Li, Bing Yin, Jingbo Shang, and Julian J. McAuley. MEMORYLLM: Towards self-updatable large language models. In *Forty-first International Conference on Machine Learning, ICML 2024*, 2024b. URL https://openreview.net/forum?id=p0lKWzdikQ.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W White, Doug Burger, and Chi Wang. Autogen: Enabling next-gen LLM applications via multi-agent conversations. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=BAakY1hNKS.

Chaojun Xiao, Pengle Zhang, Xu Han, Guangxuan Xiao, Yankai Lin, Zhengyan Zhang, Zhiyuan Liu, and Maosong Sun. InfLLM: Training-free long-context extrapolation for LLMs with an efficient context memory. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=bTHFrqhASY.

Sibo Xiao, Zixin Lin, Wenyang Gao, Hui Chen, and Yue Zhang. Long context scaling: Divide and conquer via multi-agent question-driven collaboration, 2025. URL https://arxiv.org/abs/2505.20625.

Kai Xiong, Xiao Ding, Yixin Cao, Ting Liu, and Bing Qin. Examining inter-consistency of large language models collaboration: An in-depth analysis via debate. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023. URL https://aclanthology.org/2023.findings-emnlp.508/.

Zhenran Xu, Senbao Shi, Baotian Hu, Jindi Yu, Dongfang Li, Min Zhang, and Yuxiang Wu. Towards reasoning in large language models via multi-agent peer review collaboration, 2023. URL https://arxiv.org/abs/2311.08152.

Yuchen Yan, Yongliang Shen, Yang Liu, Jin Jiang, Mengdi Zhang, Jian Shao, and Yueting Zhuang. Inftythink: Breaking the length limits of long-context reasoning in large language models, 2025. URL https://arxiv.org/abs/2503.06692.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: deliberate problem solving with large language models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, 2023. URL https://dl.acm.org/doi/abs/10.5555/3666122.3666639.

Howard Yen, Tianyu Gao, Minmin Hou, Ke Ding, Daniel Fleischer, Peter Izsak, Moshe Wasserblat, and Danqi Chen. HELMET: How to evaluate long-context models effectively and thoroughly. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=293V3bJbmE.

Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, and Hao Zhou. Memagent: Reshaping long-context llm with multi-conv rl-based memory agent, 2025. URL https://arxiv.org/abs/2507.02259.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Hao, Xu Han, Zhen Thai, Shuo Wang, Zhiyuan Liu, and Maosong Sun. ∞Bench: Extending long context evaluation beyond 100K tokens. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024a. URL https://aclanthology.org/2024.acl-long.814.

Yusen Zhang, Ruoxi Sun, Yanfei Chen, Tomas Pfister, Rui Zhang, and Sercan O Arik. Chain of agents: Large language models collaborating on long-context tasks. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024b. URL https://openreview.net/forum?id=LuCLf4BJsr.

Jun Zhao, Can Zu, Xu Hao, Yi Lu, Wei He, Yiwen Ding, Tao Gui, Qi Zhang, and Xuanjing Huang. 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*, 2024. URL https://aclanthology.org/2024.emnlp-main.912/.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: enhancing large language models with long-term memory. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, 2024. URL https://ojs.aaai.org/index.php/AAAI/article/view/29946.

Pei Zhou, Jay Pujara, Xiang Ren, Xinyun Chen, Heng-Tze Cheng, Quoc V Le, Ed H. Chi, Denny Zhou, Swaroop Mishra, and Steven Zheng. SELF-DISCOVER: Large language models self-compose reasoning structures. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=BROvXhmzYK.

A Appendix

A.1 Failure Modes of CoA

CoA [Zhang et al., 2024b] exhibits two kinds of major failure modes:-

1. **Faulty Fact Extraction:** CoA summaries can be very hyper-focused on the question at hand. Because of this, CoA can fail to gather important facts if they are not immediately relevant to the query. In longer chunk sequences, CoA can fail to gather crucial evidence even when it is relevant to the query. These errors are most evident in Appendix sections A.1.1 and A.1.3
2. **Dropping Facts during the sequence:** In CoA, each worker agent is asked to generate a new summary based on the current chunk and the accumulated summary. Hence, on longer chunk sequences, it is possible for important facts to be dropped when generating the next summary. This can result in important facts whose relevance is established in later chunks being dropped, leading to wrong answers. An example can be seen in Appendix A.1.2.

This behaviour can be further exacerbated by the tendency of models to be very conservative when generating the CoA summaries, leading to very short or overly terse summary and ineffective utilization of the provided token budget.

A.1.1 Failing to Extract Broad Information

This error can be most clearly seen in the example used in Figure 1. The question asks where did *Kiara* and *Carter* meet before becoming roommates in Nigeria. To answer this question, the system has to gather the fact that **Kiara met a pale young gentleman at Miss Kiley's house; they had a fistfight in the garden** from Chunk c_1 where the identity of the *pale, young gentleman* is a mystery. This mystery is later resolved in a later Chunk c_R, where it is revealed that **Carter is the pale, young gentleman**.

Figure 1 showcases that COSMIR is able to store such information in the memory. In CoA, the summary produced does not contain any mention of the meeting with the *pale, young gentleman*. The exact summary for the first chunk is as follows:

```
Summary:

Kiara is an orphan raised by her much older sister, Mrs. Stevie Bria, and
   her kind-hearted brother-in-law, Stevie, a blacksmith, in a rural
  marsh country. As a child, Kiara has a frightening encounter with an
  escaped convict on the marshes, whom she secretly helps by stealing
  food and a file from her home. Kiara grows up feeling common and
  ashamed of her coarse hands and boots, especially after being exposed
  to the wealthy and eccentric Miss Kiley and her beautiful but haughty
  ward, Ernest. Kiara is apprenticed to Stevie as a blacksmith, but longs
   for a different, more refined life, partly due to his infatuation with
   Ernest. Throughout her childhood and apprenticeship, Kiara is close to
   Robyn, a kind and clever girl who becomes part of their household.
It was established that Kiara later becomes roommates with Carter in
  Nigeria, and that Kiara had met Carter once before. However, in the
  current source text, there is no mention of Carter or Nigeria; the
  narrative is focused on Kiara's early life in England, her family,
  apprenticeship, and formative experiences with Miss Kiley, Ernest, and
  Robyn.
Therefore, Kiara, originally from a rural English marshland where she was
   raised by her sister and apprenticed to a blacksmith, eventually
  becomes roommates with Carter in Nigeria. Kiara and Carter had met once
   before, though the details of that meeting are not provided in the
  current text. The current text provides background on Kiara's
  upbringing, character development, and relationships, but does not yet
  describe the circumstances of her meeting with Carter or her move to
  Nigeria.
```

Listing 1: Failure of CoA in gathering broad information

A.1.2 Dropping facts during Sequential Processing

The problem of CoA dropping facts when processing a sequence of chunks can be seen using the same question in Figure 1. Section A.1.1 showcases an instance of CoA not capturing relevant facts to begin with. But, across multiple CoA runs over this example, it is possible for the fact, **Kiara met a pale young gentleman at Miss Kiley's house; they had a fistfight in the garden**, to be gathered from Chunk c_1. But because the relevance of the fact is not immediately established, the fact can be dropped in the summaries of subsequent chunks which results in an irrecoverable loss of information. This can be clearly seen when we look at the CoA summary generated after Chunk c_1 and the summary generated after Chunk c_2

```
**Summary of the Source Text and Previous Context:**
The narrative follows Kiara, a young boy raised by his sister Mrs. Stevie
   Bria, a harsh and controlling woman, and her husband Stevie, a kind
  but physically strong blacksmith. Kiara s life is marked by fear and
  secrecy, as he is coerced into aiding a mysterious convict with a file
  and food, under threat of violence. The convict, later revealed to be a
   fugitive from the Hulks (prison ships), is eventually recaptured
  during a village-wide search. Kiara s interactions with the convict
  and his sister s harsh treatment shape his sense of self-worth,
  leading to feelings of inadequacy and a desire to escape his humble
  life.

Kiara is apprenticed to Stevie, but his aspirations for a better life are
   stifled by his sister s domineering nature and the oppressive
  environment of their home. Later, he visits Miss Kiley, a reclusive and
   wealthy woman, and her attendant Ernest, who treat him with disdain.
  Kiara s attempts to improve himself through education and a desire to
   become a gentleman are met with confusion and rejection, particularly
  from Ernest. There, he comes across a pale, young gentleman who fights
  him and knocks him out. A subplot involves Robyn, a kind and observant
  girl who becomes part of their household, offering Kiara a sense of
  camaraderie and practical support.

There is no mention of Carter or Nigeria in the current source text.
  Kiara and Carter have met once before but the details of that meeting
  have not been provided in the current source text.
```

Listing 2: CoA Summary for Chunk c_1

```
*Summary of the Source Text and Previous Context:**
The narrative follows Kiara, a young blacksmith s apprentice raised by
  his sister Mrs. Stevie Bria and her husband Stevie, a kind but
  physically imposing blacksmith. Kiara s life is marked by fear and
  secrecy, as he is coerced into aiding a mysterious convict with a file
  and food under threat of violence. The convict, later revealed to be a
  fugitive from the Hulks (prison ships), is recaptured during a village-
  wide search. Kiara s interactions with the convict and his sister s
   harsh treatment shape his sense of inadequacy and desire for escape.

Kiara is apprenticed to Stevie but feels stifled by his sister s
  domineering nature and the oppressive environment of their home. As
  Kiara matures, he becomes increasingly dissatisfied with his
  apprenticeship at the forge and dreams of a better life. His
  aspirations for a better life are complicated by his sister s disdain
   and the influence of Miss Kiley, a reclusive and wealthy woman, and
  her attendant Ernest, who treat him with condescension. Kiara s
  journey to self-improvement is further complicated by his growing
  feelings for Ernest and the mysterious benefactor, Mr. Dilan, who tells
   Kiara that he has "great expectations" and that he is to be brought up
   as a gentleman in Nigeria.

There is no mention of Carter in the current source text. The current
  source text sets up the stage for Kiara to move to Nigeria to become a
```

```
gentleman but it does not provide any information about Carter and
their meeting with Kiara.
```

Listing 3: CoA Summary for Chunk c_2

As can be seen in listing 2, the summary generated for Chunk c_1 contains the information about the meeting with the *pale, young gentleman*. But this information is dropped in the summary for chunk c_2 as can be seen in listing 3.

A.1.3 Failure in Extracting Specific Information

This example showcases a simple error in fact extraction where CoA fails to gather specifics related to an event, instead opting to note broad strokes information like narrative context and thematic throughlines. The query asks **How did Marianne, a character in the context, die?**. The gold answer notes that Marianne dies by **trying to jump over a fence on a horse**. The death of this character happens in the latter half of the book, in Chunk c_6. The specific passage showcasing the moment of the character's demise from the context is as follows:

```
And then we seen him way down at the end of the pasture. He must have
  jumped the fence right over there. And he come ridin  hell-for-
  leather up the hill, singin  at the top of his voice like he didn t
   have a care in the world. I didn t know your pa had such a voice. He
   was singin   Peg in a Low-backed Car  and beatin  the horse
  with his hat and the horse was goin  like mad. He didn t draw rein
  when he come near the top and we seen he was goin  to jump the
  pasture fence and we hopped up, scared to death, and then he yelled:
   Look , Lucian! Watch me take this one!   But the horse stopped right
   on his haunches at the fence and wouldn t take the jump and your pa
  went right over his head. He didn t suffer none. He was dead time we
  got to him. I guess it broke his neck.
```

Listing 4: Passage from the Context for the Question

This incident is wrapped in a broader context of Marianne grieving the death of his wife, Lucian and the emotional and physical toll of the post civil war era. Given this context, CoA is able to note the broader context about Marianne's grief and post war difficulties but CoA omits the more direct information related to how the character died. The exact summary is as follows:

```
**Summary of the Source Text and Previous Context:**
The narrative follows **Debbie O Kerry Kaitlin**, a Southern woman
  navigating the challenges of post-Civil War Reconstruction, personal
  ambition, and familial tensions. Earlier summaries indicated that her
  father, **Marianne O Kerry**, an Irish-born plantation owner, was
  alive and enduring the war s aftermath, struggling to manage Tara s
   estate while maintaining his pride and resilience. However, in the **
  current source text** (Chapter XXXIX), it is revealed that **Marianne
  O Kerry dies in June**, shortly after Debbie s return to Jonesboro.
   This is confirmed through a message from **Will Blain** and later
  elaborated by **Will** during the funeral, where he describes
  Marianne s death as a result of his **broken heart following the
  death of his wife, Lucian O Kerry**, and his inability to cope with
  the emotional and physical toll of the post-war era. The text
  explicitly states that Marianne s death occurs **before Debbie s
  return to Atlanta**, and his passing is marked by a **funeral at Tara
  **, where his **legacy as a Southern gentleman and fighter for his land
  ** is acknowledged. The earlier summary likely omitted this detail, but
   the **current text confirms Marianne s death in Chapter XXXIX** due
  to grief and the hardships of Reconstruction.

**Answer to the Query:**
In the **current source text** (Chapter XXXIX), **Marianne O Kerry dies
   in June** after the Civil War, primarily due to **grief over the death
   of his wife, Lucian**, and the **emotional and physical toll of
  Reconstruction**. His death is confirmed via a message from **Will
```

```
Blain** to Debbie, who returns to Jonesboro to attend the funeral.
During the eulogy, Will explains that M a r i a n n e s heart had "died"
with L u c i a n s passing, leaving him "addled" and unable to recover.
The text does not provide explicit details of his death (e.g., illness,
 accident, or natural causes), but it emphasizes his **spiritual and
emotional collapse** as the central cause. This contradicts earlier
summaries that portrayed him as alive, but the **current narrative
definitively states his death** in Chapter XXXIX.
```

Listing 5: CoA Summary for Chunk c_6

A.2 Prompts Used in COSMIR

```
You are an expert research planner. Your task is to devise an exhaustive
  research strategy to answer a complex MAIN_QUERY. The goal is not just
  to find the direct answer, but to generate a set of broad, overlapping
  "Information Nets" that will reliably catch all the necessary facts,
  even if they are indirect or their relevance is not immediately obvious
  .

### The Thought Process

Follow this three-step thinking process to generate your questions:

1.  **Deconstruct the Query:** Identify the core entities, the central
  event/relationship, and all constraints (temporal, locational, etc.).

2.  **Formulate a Multi-Pronged Strategy:** Based on the deconstruction,
  define your angles of attack.
    *   **The Direct Approach:** Formulate a question that tracks the
      direct interaction or causal link between the core components of
      the query. This is your primary target.
    *   **The Decomposed Approach (Crucial Step):** Assume the direct
      answer might be incomplete or misleading. To find the full context,
       investigate each core entity's history *independently* within the
      query's constraints. This allows you to discover the underlying
      factors and connections that explain the central event.

3.  **Generate Broad, Far-Reaching Questions:** Convert your strategy
  into a set of questions. These questions should act as directives for a
   comprehensive note-taking process.

---
### Example of the Thought Process in Action

**MAIN_QUERY:** "What was the primary reason Project 'Orion' was
  cancelled following the acquisition of 'Innovate Corp'?"

**1. Deconstruction:**
*   **Core Entities:** `Project 'Orion'`, `'Innovate Corp'`.
*   **Central Event:** `cancelled`.
*   **Constraints:** `following the acquisition` (temporal and potential
  causal link).

**2. Strategy Formulation:**
*   **Direct Approach:** I need to find the officially stated reason for
  the cancellation of 'Orion' and see how it connects to the acquisition.
*   **Decomposed Approach:** The official reason might not be the whole
  story. The real cause lies at the intersection of the two entities'
  independent histories. I must build a complete picture of both 'Orion'
  and 'Innovate Corp' leading up to the cancellation.
    *   First, I will research Project 'Orion's' history on its own. What
       were its goals, budget, progress, and known problems?
```

```
    *   Second, I will research ’Innovate Corp’. What technology did they
        possess? What was the strategic purpose of their acquisition?
    *   By understanding both entities in isolation, I can cross-
      reference the timelines to uncover the true reason for the
      cancellation (e.g., ’Innovate Corp’s’ technology made ’Orion’
      redundant, the acquisition shifted budget priorities, etc.).

**3. Generate Questions (The "Information Nets"):**
*   (From the Direct Approach) -> "Find all official statements, memos,
  or post-mortems that explicitly state the reason for Project ’Orion’s’
  cancellation."
*   (From the Decomposed Approach for ’Orion’) -> "What is the complete
  history of Project ’Orion’ *before the acquisition*: its stated goals,
  budget, key personnel, major milestones, and any documented challenges
  or internal reviews."
*   (From the Decomposed Approach for ’Innovate Corp’) -> "what is the
  core technology and product line of ’Innovate Corp’ at the time of its
  acquisition. What was the stated business strategy behind the
  acquisition?"
*   (To link the contexts) -> "What organizational changes, budget
  reallocations, or technology integrations occurred between the teams of
   Project ’Orion’ and ’Innovate Corp’ after the acquisition was
  finalized?"

**MAIN_QUERY:** "Where was the first documented contact between Norse
  voyagers and the Indigenous peoples of what is now North America?"

**1. Deconstruction:**
*   **Core Entities:** ‘Norse voyagers‘, ‘Indigenous peoples of North
  America‘.
*   **Central Event:** ‘first documented contact‘.
*   **Constraints:** ‘where‘ (location) and ‘first‘ (chronology); note
  ambiguity in what counts as "documented" (Norse texts, Indigenous oral
  history, or archaeology).

**2. Strategy Formulation:**
*   **Direct Approach:** Locate the earliest explicit records or securely
   dated artifacts that document an encounter between Norse voyagers and
  Indigenous peoples.
*   **Decomposed Approach (Two overlapping information nets):**
    *   **Net A     Norse / Euro Records & Material Evidence:** Gather
      Norse saga passages, contemporaneous chronicles, runic or other
      inscriptions, and archaeological sites with Norse artifacts in
      Atlantic/North American regions; extract dates, claimed locations,
      and any mention of locals.
    *   **Net B     Indigenous Oral Traditions & Local Archaeology:**
      Compile Indigenous oral histories, place-names, and archaeological
      reports that describe encounters with outsiders or show foreign
      artifacts or cultural change; extract dating, locality, and
      descriptions.
    *   The union of Nets A and B catches earliest "documentation"
      regardless of genre.

**3. Generate Questions (The "Information Nets"):**
*   (From the Direct Approach) -> "What is the chronologically earliest
  explicit written accounts or European chronicles claiming Norse contact
   with Indigenous peoples, with exact quotations and dates."
*   (From Net A) -> "List archaeological sites in Atlantic Canada /
  nearby with securely dated Norse artifacts; for each, describe dating
  evidence and whether Indigenous Norse interaction is evident."
*   (From Net B) -> "Collect Indigenous oral histories and regional
  archaeological reports that describe early encounters with seafaring
  outsiders, including dating and locality details."
*   (To link the contexts) -> "For each Norse-dated site or saga
  reference, is there corresponding Indigenous evidence (oral or
```

```
  archaeological) at the same place/time? For Indigenous-suggested cases,
   is there any Norse material or European mention nearby?"
*   (Edge cases) -> "Could artifacts be trade items rather than evidence
  of direct contact? How do radiocarbon and stratigraphic dates constrain
   'first' claims?"

---
### YOUR TASK

Now, apply this exact same thought process to the following MAIN_QUERY.

After thinking return this output format:
```yaml
questions:
  - "Broad Question from Direct Approach"
  - "Broad Question from Decomposed Approach (Entity 1)"
  - "Broad Question from Decomposed Approach (Entity 2)"
  # ... and so on
gathered_facts: []
inferred_facts: []
answer: ""
```
MAIN_QUERY: {{query}}
```

Listing 6: PLANNER Prompt

```
Respond with YAML format ONLY. Do not use markdown code blocks or any
  other formatting.

Extract ALL relevant facts from the CONTEXT_CHUNK that could help answer
  the MAIN_QUERY.
Pay special attention to:
- Named entities (organizations, satellites, technologies, people)
- Relationships between entities (who made what, who operated what)
- Historical connections (what came before what, experimental vs
  operational)
- Technical specifications and capabilities

Return the complete updated YAML structure with new facts added:

gathered_facts:
  - "new fact from chunk"

MAIN_QUERY: {{query}}
CONTEXT_CHUNK: {{chunk}}
CURRENT_MEMORY:
{{memory}}
```

Listing 7: EXTRACT Phase Prompt

```
Respond with YAML format ONLY. Do not use markdown code blocks or any
  other formatting.

Based on the gathered facts, make logical inferences that help answer the
   MAIN_QUERY.
Look for:
- Connections between entities mentioned in different facts
- Historical relationships (what led to what)
- Organizational relationships (who owns/operates/manufactures what)
- Timeline connections (experimental versions leading to operational
  versions)

MAIN_QUERY: {{query}}

```

```
Return the complete updated YAML structure:

inferred_facts:
  - "existing inferred facts"
  - "new logical inferences"

CURRENT_MEMORY:
{{memory}}
```

Listing 8: Infer Phase Prompt

```
Respond with YAML format ONLY. Do not use markdown code blocks or any
  other formatting.

Remove answered questions and optionally add new ones.

MAIN_QUERY: {{query}}

Return exactly this YAML structure:

questions:
  - "remaining unanswered questions or newly added questions"

CURRENT_MEMORY:
{{memory}}
```

Listing 9: Refine Sub-agent Prompt

```
Respond with YAML format ONLY. Do not use markdown code blocks or any
  other formatting.

Based on the gathered facts and inferences, answer this question: {{query
  }}

Analysis approach:
1. Identify all relevant entities mentioned in the facts
2. Trace relationships and connections between entities
3. Follow logical chains to reach the final answer
4. Provide a direct, concise answer

{{memory}}

Return exactly this YAML structure:

answer: "concise answer here"
questions: []

{TASK_SPECIFIC_INST}
```

Listing 10: Manager Prompt