MegaMem: A Retrieval Solution for Ultra-Large Context Windows

Xinyuan Song[1,2] Bowen Zhu[1] Hasibul Haque[1] Liang Zhao[1,2]
[1]Causal Dynamics Lab, USA [2]Emory University, USA

Abstract

Modern language models and agents increasingly require persistent memory for complete codebases, long interaction histories, and heterogeneous enterprise records. The key challenge is to keep hundreds of millions of tokens searchable while passing only bounded source evidence to the answer model. We introduce MegaMem, a source-resolved dual-view retrieval system that separates semantic access from generation evidence. Distilled records and detailed evidence are searched with original and transformed queries; every distilled hit resolves to an immutable source ID before reciprocal-rank fusion, deduplication, and cross-encoder reranking; and only the highest-ranked detailed evidence within a fixed budget supports generation. Post-answer attribution then identifies which loaded sources support the fixed answer. We evaluate MegaMem on EnterpriseRAG-Bench, which contains more than 500,000 heterogeneous enterprise documents and approximately 650M tokens. MegaMem improves Overall from 68.22 to 82.26 and reaches 86.50 Correctness. These results show that MegaMem supports ultra-large persistent memory while preserving strong answer accuracy under a bounded generation context. By separating searchable memory scale from answer-context size, MegaMem provides a practical path toward accurate retrieval over memories ranging from hundreds of millions to one billion tokens. Our code is available at https://github.com/xfab-xinyuansong/MegaMem.git.

1 Introduction

Modern large language models and agents increasingly depend on large context to support long-term interaction, planning, tool use, and knowledge-intensive decision making (Packer et al., 2023; Wang et al., 2023b; Zhong et al., 2023; Maharana et al., 2024; Wu et al., 2025; Chhikara et al., 2025; Kang et al., 2025). Memory management has therefore become a core problem in agent systems, since an agent must retain observations, task states, interaction histories, and prior experience across steps and sessions. In practice, modern language systems require persistent memory that can store complete codebases, long interaction histories, and heterogeneous enterprise records (Sun et al., 2026; Choubey et al., 2025; Yu et al., 2025). Native long-context models can process millions of tokens in one call (Gemini Team, 2024), while retrieval-based models can search datastores containing trillions of tokens (Borgeaud et al., 2022). Persistent-memory systems instead store information outside the active context and retrieve selected content when needed (Packer et al., 2023; Wang et al., 2023b). Recent systems further organize, update, and retrieve long-term memory to support continued interaction, personalization, and cross-session reasoning (Zhong et al., 2023; Maharana et al., 2024; Wu et al., 2025; Chhikara et al., 2025; Kang et al., 2025).

However, only a small fraction of such memory usually supports one answer. Directly loading more content creates a quality–cost tradeoff: relevant evidence may be missed because of its position (Liu et al., 2023), performance can decline as context length and document count increase (Hsieh et al., 2024; Levy et al., 2025; Du et al., 2025), and irrelevant passages can distract the model (Cuconasu et al., 2024). Processing the full repository also causes attention computation, latency, and serving cost to grow with stored history. Existing methods address this problem through hierarchical summaries (Sarthi et al., 2024), graph-based corpus representations (Edge et al., 2024; Guo

et al., 2024; Gutiérrez et al., 2024), global memory and retrieval clues (Qian et al., 2025), or external memory management (Packer et al., 2023; Wang et al., 2023b; Chhikara et al., 2025; Kang et al., 2025). These methods improve access to large stores, but compressed records may omit dates, exceptions, conditions, or conflicts needed for a correct answer. In contrast, retrieving only detailed chunks preserves exact support but increases distractor exposure and weakens retrieval precision as the corpus grows (Cuconasu et al., 2024; Levy et al., 2024, 2025).

Ultra-large memory retrieval must therefore support three goals: semantic retrieval across different expressions, source-faithful evidence for generation, and precise attribution to the sources used. We distinguish *persistent context*, the full corpus that the system can search, from *evidence context*, the bounded detailed source evidence consumed for one answer. The goal is to scale the former without increasing the latter at the same rate. We introduce MegaMem, which assigns these goals to separate stages: distilled records support retrieval, each distilled hit resolves to an immutable detailed-source identifier, only detailed source evidence enters generation, and post-answer attribution identifies the loaded sources that support the fixed answer (Hsia et al., 2025; Levy et al., 2024).

Ultra-large memory retrieval must support semantic access, source-faithful generation, and precise attribution. We distinguish *persistent context*, the full searchable corpus, from *evidence context*, the bounded detailed source evidence used for one answer. MegaMem scales the former while keeping the latter fixed: distilled records support retrieval, every distilled hit resolves to detailed source evidence through an immutable source ID, and only the highest-ranked resolved chunks enter generation. Original and transformed queries search both views, followed by reciprocal-rank fusion, deduplication, and cross-encoder reranking. As shown in Figure 1, this design separates persistent memory, retrieval candidates, loaded evidence, and reported sources, allowing MegaMem to operate over hundreds of millions of tokens without passing the full memory to the answer model.

This design builds on prior work in native long context, persistent memory, structured retrieval, multi-stage retrieval, and attribution (Gemini Team, 2024; Packer et al., 2023; Sarthi et al., 2024; Cormack et al., 2009; Gao et al., 2023), and leads to three research questions. First, does separating retrieval records from generation evidence improve end-to-end answer quality? Second, which components—dual-view retrieval, distillation, query transformation, reranking, and post-answer attribution—produce the observed gains? Third, as persistent memory grows under a fixed evidence budget, does performance decline because retrieval fails to find the required evidence or because the answer model cannot use the selected evidence?

We evaluate these questions on EnterpriseRAG-Bench, a recent benchmark for retrieval over realistic enterprise knowledge (Sun et al., 2026). It contains more than 500,000 documents from heterogeneous enterprise sources and 500 questions covering retrieval, multi-document reasoning, conflict resolution, constrained search, and missing-information cases. The full corpus used in our study contains approximately 650M tokens, providing a direct test of retrieval over ultra-large and noisy memory. On the disjoint 400-question validation split, MegaMem provides an effective solution to ultra-large memory retrieval, improving Overall from 68.22 to 82.26 and reaching 86.50 Correctness. The strongest gains come from dual-view retrieval and distillation, confirming that separating semantic access from source-faithful evidence is central to the design. Post-answer attribution further reduces the reported source set without changing answer quality, improving source reporting while preserving the final prediction. This work makes three contributions:

- We study the problem of ultra-large persistent memory at the scale of 250M to 1B tokens, where a language system must keep the full memory searchable while passing only a bounded amount of evidence to the answer model.
- We introduce MegaMem, a source-resolved dual-view retrieval system that searches both distilled records and detailed evidence with original and transformed queries, resolves all distilled hits to immutable source IDs before reciprocal-rank

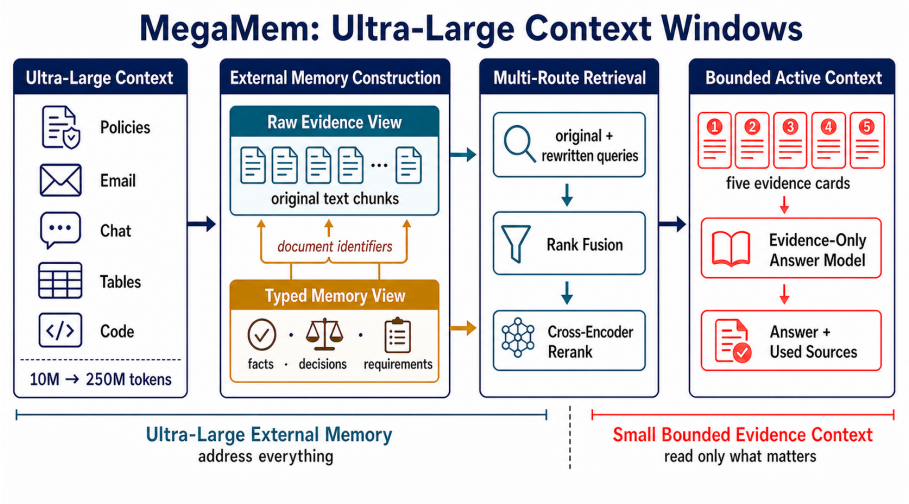

Figure 1: **MegaMem maps ultra-large persistent memory to a bounded evidence context.** Enterprise knowledge is stored as source-linked raw evidence and typed memories, retrieved through multiple query routes, fused and reranked, and then reduced to a small set of detailed evidence cards used for answer generation and source attribution.

fusion, deduplication, and cross-encoder reranking, loads only the highest-ranked detailed evidence under a fixed budget, and applies post-answer attribution to the fixed answer.

- We evaluate MegaMem against strong retrieval and memory baselines and provide controlled analyses of end-to-end quality, component effects, wall-clock cost, token use, evidence-budget trade-offs, and memory scaling. MegaMem achieves the best overall performance while keeping the generation context bounded as persistent memory grows.

2 Problem Formulation

2.1 Persistent Context and Evidence Context

Let $\mathcal{D}$ denote the full persistent corpus with T tokens. For a question q, the system retrieves an evidence set $E(q) \subseteq \mathcal{D}$ under a fixed budget B:

$$E(q) \subseteq \mathcal{D}, \qquad \mathrm{tok}(E(q)) \leq B, \qquad B \ll T. \tag{1}$$

We refer to $\mathcal{D}$ as the *persistent context* and $E(q)$ as the *evidence context*. The goal is to scale the searchable corpus while keeping the evidence context bounded, even though retrieval becomes harder as distractors increase.

2.2 Source-Resolved Retrieval

Prior work uses compressed or structured representations to improve retrieval over large corpora (Sarthi et al., 2024; Edge et al., 2024; Qian et al., 2025). In MegaMem, such representations are used only to locate relevant content: every compressed hit is resolved to its corresponding detailed source chunk before fusion, reranking, and generation. The answer is therefore produced only from detailed evidence:

$$a = \mathcal{A}\big(q, E(q)\big), \qquad E(q) \subseteq \mathcal{X}, \tag{2}$$

where $\mathcal{X}$ is the set of detailed source chunks.

3 Method

MegaMem consists of three stages: dual-view memory construction, multi-route retrieval, and bounded-evidence generation with post-answer attribution. As illustrated in Figure 1, the system separates retrieval-oriented representations from generation evidence. Distilled memories improve semantic access to the corpus, but every retrieved memory is resolved to its corresponding detailed source chunk before fusion, reranking, and context construction. Consequently, only detailed source evidence is passed to the answer model or exposed as a reported source.

3.1 Dual-View Memory Construction

Given a document collection, we first normalize each document and segment it into

section-aware evidence chunks using headings and paragraph boundaries. Long paragraphs are further split by sentence and token boundaries. This produces a detailed evidence view $\mathcal{X} = \{x_1, \ldots, x_n\}$, where each chunk retains its original content and source metadata.

For each detailed chunk x_i, an extractor produces up to three compact typed memories:

$$m_{ij} = (\tau_{ij}, u_{ij}, v_{ij}, \pi_{ij}), \qquad \pi_{ij} = i, \quad (3)$$

where τ_{ij} specifies the memory type, u_{ij} is a retrieval key, v_{ij} is a concise statement supported by x_i, and π_{ij} links the memory to its source chunk. We use five memory types: fact, procedure, definition, requirement, and decision. Chunks without useful semantic content may produce no memory.

The resulting corpus is indexed in two complementary views. The detailed index stores embeddings of the original evidence chunks together with their source metadata, while the distilled index stores embeddings of the typed memories and their source pointers. The distilled view is optimized for semantic matching, whereas the detailed view preserves the exact wording, conditions, and provenance required for answering. Both views therefore refer to the same underlying evidence rather than forming separate evidence pools.

Atomic memories are extracted with `gpt-5.4-mini` (OpenAI, 2026b). *MiniExtractor* retains only these atomic memories, whereas *FullExtractor* additionally uses `gpt-5.4` to construct higher-level retrieval abstractions (OpenAI, 2026a). Both configurations use `text-embedding-3-small` for indexing (OpenAI, 2024).

The hierarchy and relation structures explored during development are not used in the final system. As shown in Table 9, the final design retains only the detailed and distilled views. Algorithm 2 in Appendix G provides the complete construction procedure.

3.2 Multi-Route Retrieval and Evidence Resolution

Given a question q, MegaMem constructs multiple query routes from the original question, including a canonical rewrite and terminology-diverse expansions. Each route searches both the detailed index I_d and the distilled index I_m. The detailed route directly retrieves source chunks, whereas the distilled route retrieves compact memories that may better match alternative expressions of the same information need.

All distilled hits are resolved to their corresponding detailed chunks before candidate aggregation. Let $\mathrm{Retr}(r, I)$ denote the ranked set of top-k items retrieved from index I using query route r, and let $\pi(m)$ map a distilled memory m to its associated detailed source chunk. The candidate set is therefore defined over detailed evidence identities:

$$C(q) = \bigcup_{r \in \mathcal{Q}(q)} \left(\mathrm{Retr}(r, I_d) \cup \pi\left(\mathrm{Retr}(r, I_m)\right) \right), \quad (4)$$

where $\mathcal{Q}(q)$ contains the original and transformed queries, I_d is the detailed index, and I_m is the distilled index. This resolution step allows multiple retrieval routes to improve recall without treating compressed memories as independent evidence items.

Because scores from different routes are not directly comparable, we combine their rankings using weighted reciprocal rank fusion (Cormack et al., 2009):

$$S_{\mathrm{RRF}}(x \mid q) = \sum_{r \in \mathcal{L}(q)} \frac{w_r \, \mathbf{1}[x \in r]}{\gamma + \mathrm{rank}_r(x)}, \quad (5)$$

where $\mathcal{L}(q)$ is the set of ranked retrieval lists, w_r is the weight assigned to route r, and $\gamma = 60$. We assign larger weights to the original and canonical semantic routes than to auxiliary expansions.

A cross-encoder then reranks the fused candidate pool using the question and full detailed chunk content (Nogueira and Cho, 2019; Ma et al., 2023b). Duplicate candidates are merged by source identity, and the highest-ranked chunks are packed until the evidence budget B is reached. The resulting evidence context contains only detailed source chunks, while query transformation and distilled retrieval serve solely to improve candidate discovery.

3.3 Bounded Generation and Post-Answer Attribution

The retrieval stage returns a bounded set of detailed evidence chunks $E(q)$, which forms the

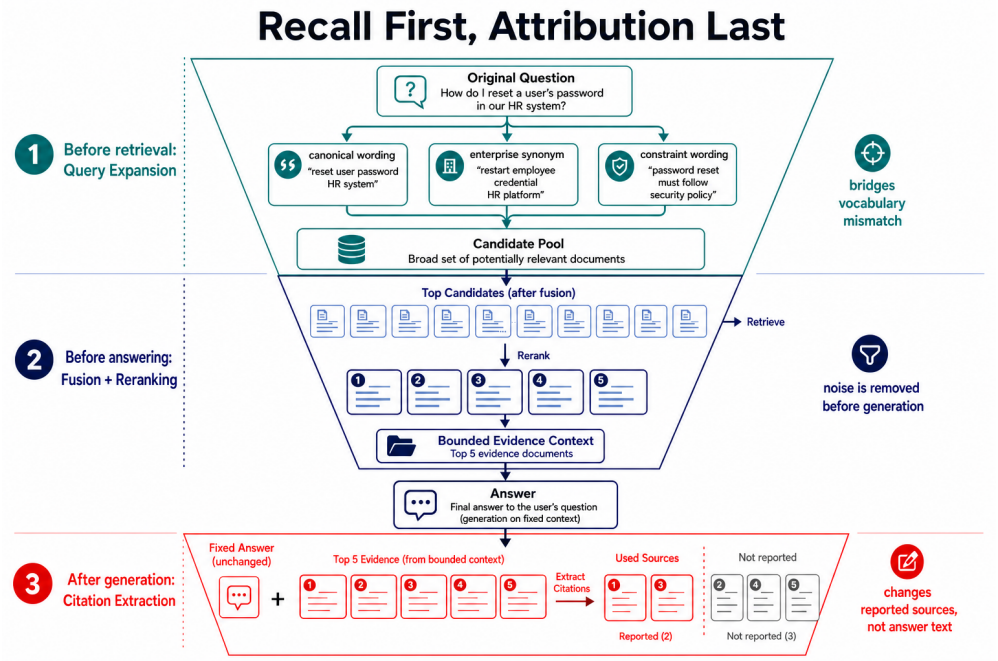

Figure 2: **Inference order in MegaMem.** Query transformation broadens retrieval, source resolution and reranking construct the bounded evidence context, and post-answer attribution filters the reported sources without modifying the answer.

Algorithm 1 Multi-route retrieval with source resolution

Require: question q, detailed index I_d, distilled index I_m, evidence budget B
$\mathcal{Q} \leftarrow \textsc{Transform}(q) \cup \{q\}$
retrieve ranked candidates from I_d and I_m for each query in $\mathcal{Q}$
resolve every distilled candidate m to its detailed source chunk $\pi(m)$
fuse resolved rankings with weighted reciprocal rank fusion
rerank and deduplicate candidates by detailed source identity
$E(q) \leftarrow \textsc{Pack}(\text{ranked detailed chunks}, B)$
return $E(q)$

complete context available to the answer model. The answer is generated under an evidence-only instruction:

$$a = \mathcal{A}\big(q, E(q)\big). \tag{6}$$

No distilled memory or unselected source is exposed during generation.

After the answer is fixed, a separate attribution module identifies which retrieved documents directly support its claims:

$$Z = \mathcal{C}\big(a, E(q)\big), \qquad Z \subseteq \mathrm{docs}\big(E(q)\big). \tag{7}$$

Because attribution is performed after generation, it can only remove unsupported or unused sources from the reported set; it cannot revise the answer or introduce evidence that was not shown to the answer model.

This ordering separates three roles in the pipeline. Query transformation broadens candidate recall, reranking selects the evidence used for generation, and post-answer attribution reduces the final source set. In our ablation, removing attribution leaves answer quality unchanged but increases the average number of reported documents from 2.31 to 5.00, confirming that the module primarily improves source precision rather than answer correctness. The complete runtime procedure is provided in Figure 2 and Algorithm 3 in Appendix G (Gao et al., 2023; Qi et al., 2024).

4 Experimental Setup

Benchmarks and protocols. We primarily evaluate MegaMem on EnterpriseRAG-Bench, which contains 500 questions over more than 500,000 heterogeneous enterprise documents (Sun et al., 2026). We use 100 questions for architecture, prompt, and hyperparameter development, then freeze the system and evaluate on the disjoint 400-question validation split. Table 1 summarizes the main analyses. Because the question sets, memory scales, and extrac-

tors differ across protocols, results should be compared only within matched settings. We additionally evaluate transfer on FinanceBench, HotpotQA, LoCoMo, and UltraDomain (Islam et al., 2023; Yang et al., 2018; Maharana et al., 2024; Qian et al., 2025).

Baselines and metrics. We compare against lexical and dense retrieval baselines, including BM25, DPR, and Contriever (Robertson and Zaragoza, 2009; Karpukhin et al., 2020; Izacard et al., 2022), as well as hierarchical, graph-based, and memory-guided retrieval methods (Sarthi et al., 2024; Edge et al., 2024; Guo et al., 2024; Qian et al., 2025). We adopt the evaluation metrics defined in EnterpriseRAG-Bench (Sun et al., 2026). *Correctness* measures whether the generated answer reaches the correct conclusion while preserving the required scope, conditions, and conflicts. *Completeness* measures how fully the answer covers the facts required by the reference answer. *Overall* is the benchmark's aggregate answer-quality score derived from Correctness and Completeness. *Document Recall* measures the fraction of gold source documents recovered by the system. *InvDocs* measures the fraction of reported documents judged invalid or unsupported, with lower values indicating better source precision. Atomic extraction uses `gpt-5.4-mini` (OpenAI, 2026b); higher-level abstraction, query transformation, answering, attribution, and evaluation use `gpt-5.4` (OpenAI, 2026a); and all dense indices use `text-embedding-3-small` (OpenAI, 2024). Appendix B provides the full protocol and implementation details.

5 Results

We first present the main scaling results, showing that MegaMem remains effective as persistent memory grows to hundreds of millions of tokens under a fixed evidence budget. We then compare MegaMem with direct retrieval baselines and analyze the contribution of individual components. Together, these experiments demonstrate that MegaMem provides a strong and scalable solution for retrieval over ultra-large persistent memory. Because the protocols differ in question set, memory scale, and extractor, we interpret each result within its own experimental setting.

5.1 Scaling to Ultra-Large Persistent Memory

We next evaluate whether MegaMem remains effective as the persistent corpus grows from 20M to 250M tokens while the retrieval and evidence budgets remain fixed. As shown in Table 2 and Figure 3, MegaMem continues to produce useful answers throughout this range, achieving an Overall score of 58.02 and a Correctness score of 73.50 even at 250M tokens. This demonstrates that the system can keep hundreds of millions of tokens searchable without increasing the generation context.

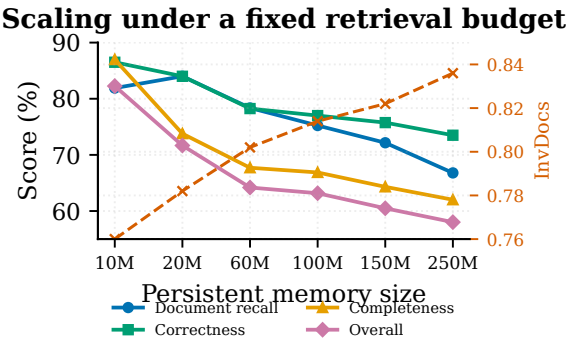

Figure 3: **Performance as persistent memory grows under a fixed evidence budget.** MegaMem remains operational from 20M to 250M tokens without increasing the context provided to the answer model.

Although retrieval becomes more difficult as the corpus grows, performance degrades gradually rather than collapsing. From 20M to 250M tokens, Correctness remains above 73%, while Document Recall remains above 66%. These results support the central claim of MegaMem: ultra-large persistent memory can be made addressable while keeping the evidence context bounded.

5.2 Comparison with Direct Retrieval Baselines

Table 3 compares MegaMem with standard retrieval pipelines that directly retrieve evidence from the original corpus, including lexical and single-vector dense RAG (Robertson and Zaragoza, 2009; Karpukhin et al., 2020; Izacard et al., 2022). We also include compressed-memory variants, the published EnterpriseRAG-Bench leader, and intermediate MegaMem configurations. All systems are evaluated on the same 400-question validation

Table 1: **Main EnterpriseRAG-Bench protocols.** Results are comparable only within matched settings.

Analysis	Questions	Memory scale	Extractor	Purpose
Headline comparison	Validation, 400	10M	FullExtractor	End-to-end quality
Scaling trace	Validation, 400	20M–250M	MiniExtractor	Scale behavior
Component ablation	Full benchmark, 500	10M	FullExtractor	Component diagnosis
Gold intervention	Validation, 400	20M/60M/250M	Both	Failure localization

Table 2: **EnterpriseRAG-Bench performance across persistent-memory scales.** All rows use the 400-question validation split, MiniExtractor, GPT-5.4 answering, and fixed retrieval and evidence budgets. InvDocs is in $[0, 1]$, and lower is better.

Memory scale	Overall	Document Recall	Correctness	Completeness	InvDocs
20M	71.67	84.00	84.00	73.78	0.782
60M	64.19	78.36	78.25	67.71	0.802
100M	63.17	75.23	77.00	66.87	0.814
150M	60.49	72.16	75.75	64.31	0.822
250M	58.02	66.79	73.50	62.01	0.836

split at the 10M memory scale.

MegaMem achieves the best Overall, Correctness, Completeness, and Document Recall. Relative to the published benchmark leader, the final system improves Overall by 20.58%, Correctness by 6.00%, Completeness by 19.38%, and Document Recall by 3.64%. These gains show that combining distilled retrieval keys with source-resolved detailed evidence is more effective than directly retrieving only from the original corpus.

Post-answer attribution further improves source reporting, reducing InvDocs from 0.774 to 0.760 without changing answer quality. This confirms that attribution removes unused or unsupported sources while preserving the generated answer.

5.3 Component Analysis

Table 4 reports a leave-one-component-out study on all 500 EnterpriseRAG-Bench questions at the 10M memory scale. The answer model, embeddings, evaluator, candidate budget, and evidence budget are fixed across all runs; each row removes only the specified component from the full MegaMem pipeline.

The dual index and distillation provide the largest gains. Removing the dual index reduces Overall by 17.91% and Document Recall from 81.90% to 70.20%, while removing distillation reduces Overall by 14.15%. Query expansion and reranking provide additional improvements of 3.55% and 4.97%, respectively. These results show that complementary retrieval representations determine the quality of the candidate set, while query transformation and reranking refine the final evidence selection.

Post-answer attribution serves a different role. Removing it leaves all answer-quality metrics unchanged but increases the average number of reported documents from 2.31 to 5.00. Thus, attribution removes 53.8% of the retrieved source set without changing the generated answer. Query expansion also recovers 13 of 33 targets missed by the base retriever, further confirming its contribution to recall.

5.4 Efficiency and Transfer

External transfer. Table 5 evaluates whether MegaMem transfers beyond EnterpriseRAG-Bench. On FinanceBench (Islam et al., 2023) and HotpotQA (Yang et al., 2018), MegaMem achieves Overall scores of 81.05 and 86.37, demonstrating strong performance on financial question answering and multi-hop reasoning. LoCoMo (Maharana et al., 2024) and UltraDomain (Qian et al., 2025) are more challenging because they require retrieval over broader shared stores with temporal dependencies, cross-session evidence, and cross-domain distractors. Although performance is lower in these settings, the results identify clear directions for extending the current document-oriented memory representation.

Table 3: **Comparison with direct retrieval baselines on EnterpriseRAG-Bench.** All rows use the 400-question validation split at the 10M persistent-memory scale. Lexical RAG and single-vector dense RAG retrieve evidence directly from the original corpus, while MegaMem uses source-resolved dual-view retrieval. InvDocs is in $[0, 1]$, and lower is better. A dash indicates that the baseline does not produce the benchmark-compatible reported-document set required to compute InvDocs.

System	Overall	Correctness	Completeness	Document Recall	InvDocs
Lexical RAG	40.27	52.75	46.95	83.71	–
Single-vector dense RAG	43.03	55.50	51.08	83.74	–
Hierarchy-only memory	60.48	73.00	64.19	68.59	0.818
Relation-only memory	59.16	72.50	63.02	66.74	0.828
Published benchmark leader	68.22	81.60	72.86	79.02	0.470
Reranked dual view	74.46	84.25	77.58	75.78	0.804
FullExtractor + Expansion	82.26	86.50	86.98	81.90	0.774
FullExtractor + Expansion + Attribution	82.26	86.50	86.98	81.90	0.760

Table 4: **Leave-one-component-out ablation on EnterpriseRAG-Bench.** All rows use the full 500-question benchmark, 10M persistent memory, FullExtractor, GPT-5.4 answering and evaluation, and fixed candidate and evidence budgets. Change is relative to the full configuration.

Configuration	Overall	Change (%)	Correctness	Document Recall	Completeness	Reported Documents
Full configuration	82.26	–	86.50	81.90	86.98	2.31
No query expansion	79.34	-3.55	83.25	81.90	84.58	2.31
No reranker	78.17	-4.97	82.75	81.90	83.74	2.31
No distillation	70.62	-14.15	74.75	73.10	77.64	2.31
No dual index	67.53	-17.91	72.50	70.20	75.18	2.31
No post-answer attribution	82.26	0.00	86.50	81.90	86.98	5.00

Table 5: **Cross-dataset transfer results.** FullExtractor and GPT-5.4 are evaluated on each benchmark using its stated question count and task-specific memory setting. N/A indicates that the benchmark does not define the corresponding document metric. InvDocs is in $[0, 1]$, and lower is better.

Benchmark	Questions	Correctness	Completeness	Overall	Document Recall	InvDocs
FinanceBench	150	82.00	87.29	81.05	N/A	N/A
HotpotQA	200	87.00	87.09	86.37	N/A	N/A
LoCoMo	200	38.00	42.49	42.34	52.98	0.879
UltraDomain	200	9.50	13.28	17.73	45.00	0.910

Evidence efficiency. We next compare three evidence policies while fixing candidate depth at 20 and sampling 100 questions at each memory scale. As shown in Table 6, MegaMem retains most of the accuracy of detailed-evidence-only generation while using substantially fewer input tokens. At 10M and 20M, selective detail reduces answer input by 68.6% and 68.8%, respectively, with only a 1.5-point reduction in Correctness. In contrast, distilled-only evidence is much shorter but loses most of the information required for accurate generation.

These results show that MegaMem provides a favorable accuracy–context trade-off: it reduces answer-model input by more than two thirds while preserving nearly all of the Correctness obtained with full detailed evidence. This supports the central design choice of using compressed memories for retrieval and loading detailed evidence only when needed.

Operational efficiency. The bounded evidence context keeps online inference efficient as persistent memory grows. Building the memory remains an offline cost: constructing 60M tokens requires 9.3 hours, while 250M tokens requires approximately 5.6 days. Retrieval takes only 0.30–1.10 seconds at 60M–100M, end-to-end answering takes 2.5–3.3 seconds, and the complete 500-question diagnostic costs $3.75, or $0.0075 per question.

Answerability-aware expansion. Query expansion substantially improves retrieval for answerable questions, producing relative gains of 20.8–100.0% across project, high-level, constrained, conflict, and semantic categories.

Table 6: **Evidence-efficiency comparison.** Each row uses 100 sampled questions, candidate depth 20, GPT-5.4 answering, and the stated persistent-memory scale. Token change is measured relative to detailed evidence only.

Memory scale	Evidence policy	Answer tokens/query	Token change	Correctness
10M	Detailed evidence only	6,532	0.0%	88.00
10M	Selective detail (MegaMem)	2,049	−68.6%	86.50
10M	Distilled only	1,013	−84.5%	21.00
20M	Detailed evidence only	6,497	0.0%	85.50
20M	Selective detail (MegaMem)	2,025	−68.8%	84.00
20M	Distilled only	1,020	−84.3%	17.00

However, it reduces information-not-found Correctness from 100.0% to 68.8%. A selective oracle preserves the gains on answerable categories while restoring information-not-found Correctness to 100.0%, indicating that a calibrated answerability gate is the main remaining requirement.

6 Conclusion

We introduced MegaMem, a retrieval solution for ultra-large persistent memory. By using distilled memories for semantic access and resolving all retrieved content to detailed source evidence, MegaMem keeps hundreds of millions of tokens searchable while maintaining a bounded generation context. Experiments on EnterpriseRAG-Bench show that this design improves retrieval and answering quality while remaining effective at scales up to 250M tokens.

Limitations

Most reported configurations are based on single runs, and the detailed-evidence-only packing result is estimated from a retained trace rather than a new controlled rerun. Source precision remains lower than the published EnterpriseRAG-Bench leader, and performance on LoCoMo and UltraDomain shows that the current document-oriented retrieval design does not transfer uniformly to conversational or broad cross-domain memory stores. Finally, MegaMem increases the amount of context that can be searched, but it does not extend the native attention window of the underlying language model.

References

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024. LongBench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. *arXiv preprint arXiv:2412.15204*.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, and 9 others. 2022. Improving language models by retrieving from trillions of tokens. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*.

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

Prafulla Kumar Choubey, Xiangyu Peng, Shilpa Bhagavath, Kung-Hsiang Huang, Caiming Xiong, and Chien-Sheng Wu. 2025. Benchmarking deep search over heterogeneous enterprise data. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 501–517.

Chroma. 2026a. Chroma python client reference. `https://docs.trychroma.com/reference/python/client`. Accessed 2026-08-02.

Chroma. 2026b. Managing chroma collections. `https://docs.trychroma.com/docs/collections/manage-collections`. Accessed 2026-08-02.

Yung-Sung Chuang, Wei Fang, Shang-Wen Li, Wentau Yih, and James Glass. 2023. Expand, rerank, and retrieve: Query reranking for open-domain question answering. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12131–12147.

Gordon V. Cormack, Charles L. A. Clarke, and Stefan Büttcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 758–759.

Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. 2024. The power of noise: Redefining retrieval for RAG systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Yufeng Du, Minyang Tian, Srikanth Ronanki, Subendhu Rongali, Sravan Babu Bodapati, Aram Galstyan, Azton Wells, Roy Schwartz, Eliu A. Huerta, and Hao Peng. 2025. Context length alone hurts LLM performance despite perfect retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 23281–23298.

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. *Preprint*, arXiv:2404.16130.

Shahul Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. 2024. Ragas: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (EACL): System Demonstrations*.

Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2021. SPLADE v2: Sparse lexical and expansion model for information retrieval. *arXiv preprint arXiv:2109.10086*.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6465–6488, Singapore. Association for Computational Linguistics.

Gemini Team. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. 2024. Lightrag: Simple and fast retrieval-augmented generation. *Preprint*, arXiv:2410.05779.

Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. 2024. Hipporag: Neurobiologically inspired long-term memory for large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Jennifer Hsia, Afreen Shaikh, Zora Zhiruo Wang, and Graham Neubig. 2025. RAGGED: Towards informed design of scalable and stable RAG systems. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 24139–24155.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. 2024. RULER: What's the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*.

Piotr Indyk and Haike Xu. 2023. Worst-case performance of popular approximate nearest neighbor search implementations: Guarantees and limitations. In *Advances in Neural Information Processing Systems*, volume 36.

Pranab Islam, Anand Kannappan, Douwe Kiela, Rebecca Qian, Nino Scherrer, and Bertie Vidgen. 2023. Financebench: A new benchmark for financial question answering. *arXiv preprint arXiv:2311.11944*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research (TMLR)*.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research (JMLR)*.

Suhas Jayaram Subramanya, Fnu Devvrit, Harsha Vardhan Simhadri, Ravishankar Krishnawamy, and Rohan Kadekodi. 2019. DiskANN: Fast accurate billion-point nearest neighbor search on a single node. In *Advances in Neural Information Processing Systems*, volume 32.

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, Ye Qi, and Zhicheng Dou. 2025. Hierarchical document refinement for long-context retrieval-augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*.

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. 2025. Memory OS of AI agent. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering.

In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Quinn Leng, Jacob Portes, Sam Havens, Matei Zaharia, and Michael Carbin. 2024. Long context RAG performance of large language models. *Preprint*, arXiv:2411.03538.

Mosh Levy, Alon Jacoby, and Yoav Goldberg. 2024. Same task, more tokens: the impact of input length on the reasoning performance of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*.

Shahar Levy, Nir Mazor, Lihi Shalmon, Michael Hassid, and Gabriel Stanovsky. 2025. More documents, same length: Isolating the challenge of multiple documents in RAG. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 19539–19547.

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. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Lingyuan Liu and Mengxiang Zhang. 2025. Exp4Fuse: A rank fusion framework for enhanced sparse retrieval using large language model-based query expansion. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 163–173.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics (TACL)*.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023a. Query rewriting for retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. 2023b. Fine-tuning LLaMA for multi-stage text retrieval. *arXiv preprint arXiv:2310.08319*.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 2024. Evaluating very long-term conversational memory of llm agents. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2021. Generation-augmented retrieval for open-domain question answering. *arXiv preprint arXiv:2009.08553*.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with BERT. *arXiv preprint arXiv:1901.04085*.

Rodrigo Nogueira, Zhiying Jiang, and Jimmy Lin. 2020. Document ranking with a pretrained sequence-to-sequence model. *arXiv preprint arXiv:2003.06713*.

OpenAI. 2024. New embedding models and API updates. https://openai.com/index/new-embedding-models-and-api-updates/. Accessed 2026-08-02.

OpenAI. 2026a. GPT-5.4 model. https://developers.openai.com/api/docs/models/gpt-5.4. Accessed 2026-08-02.

OpenAI. 2026b. Introducing GPT-5.4 mini and nano. https://openai.com/index/introducing-gpt-5-4-mini-and-nano/. Accessed 2026-08-02.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. 2023. Memgpt: Towards llms as operating systems. *Preprint*, arXiv:2310.08560.

Jirui Qi, Gabriele Sarti, Raquel Fernández, and Arianna Bisazza. 2024. Model internals-based answer attribution for trustworthy retrieval-augmented generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6037–6053.

Hongjin Qian, Zheng Liu, Peitian Zhang, Kelong Mao, Defu Lian, Zhicheng Dou, and Tiejun Huang. 2025. Memorag: Boosting long context processing with global memory-enhanced retrieval augmentation. In *Proceedings of the ACM on Web Conference 2025*, pages 2366–2377.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North*

American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT).

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. 2024. Raptor: Recursive abstractive processing for tree-organized retrieval. In *International Conference on Learning Representations (ICLR)*.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2024. REPLUG: Retrieval-augmented black-box language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Yuhong Sun, Joachim Rahmfeld, Chris Weaver, Weijia Chen, Roshan Desai, Wenxi Huang, and Mark H. Butler. 2026. Enterpriserag-bench: A rag benchmark for company internal knowledge. *Preprint*, arXiv:2605.05253.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogenous benchmark for zero-shot evaluation of information retrieval models. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*.

Liang Wang, Nan Yang, and Furu Wei. 2023a. Query2doc: Query expansion with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Weizhi Wang, Li Dong, Hao Cheng, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. 2023b. Augmenting language models with long-term memory. *arXiv preprint arXiv:2306.07174*.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. 2025. Longmemeval: Benchmarking chat assistants on long-term interactive memory. In *International Conference on Learning Representations (ICLR)*.

Peng Xu, Wei Ping, Xianchao Wu, Lawrence McAfee, Chen Zhu, Zihan Liu, Sandeep Subramanian, Evelina Bakhturina, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Retrieval meets long context large language models. In *The Twelfth International Conference on Learning Representations*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Tan Yu, Anbang Xu, and Rama Akkiraju. 2024. In defense of RAG in the era of long-context language models. *Preprint*, arXiv:2409.01666.

Tan Yu, Wenfei Zhou, Lei Yang, Aaditya Shukla, Meenakshi Madugula, Pritam Gundecha, Nick Burnett, Anbang Xu, Vishal Seth, Tamar Bar, Rama Akkiraju, and Vivienne Zhang. 2025. EKRAG: Benchmark RAG for enterprise knowledge question answering. In *Proceedings of the Fourth Workshop on Knowledge Augmented Methods for Natural Language Processing*.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Khai Hao, Xu Han, Zhen Leng Thai, Shuo Wang, Zhiyuan Liu, and Maosong Sun. 2024. ∞bench: Extending long context evaluation beyond 100k tokens. *arXiv preprint arXiv:2402.13718*.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. 2023. MemoryBank: Enhancing large language models with long-term memory. *arXiv preprint arXiv:2305.10250*.

A Related Work

Long context and persistent memory. Recent language models support million-token context windows (Gemini Team, 2024), yet benchmarks such as ∞Bench, RULER, and LongBench v2 show that effective context utilization remains strongly task-dependent (Zhang et al., 2024; Hsieh et al., 2024; Bai et al., 2024). Position bias, distractors, document multiplicity, and increasing input length can degrade performance well before the nominal context limit (Liu et al., 2023; Levy et al., 2024; Cuconasu et al., 2024; Levy et al., 2025; Du et al., 2025). Retrieval-augmented language models instead access external non-parametric memory at inference time (Lewis et al., 2020; Borgeaud et al., 2022; Izacard et al., 2023; Shi et al., 2024). Prior comparisons show that retrieval remains useful alongside long native context windows, while LongRAG adjusts the retrieval unit to better balance retrieval and generation costs (Xu et al., 2024; Jiang et al., 2024; Leng et al., 2024; Yu et al., 2024). Persistent-memory systems extend this setting across interactions and sessions (Packer et al., 2023; Wu et al., 2025; Chhikara et al., 2025; Kang et al., 2025). MegaMem targets the same general problem at larger scale by separating the searchable persistent context from the bounded evidence context used for generation.

Structured retrieval. Prior work improves retrieval through hierarchical summaries, graph structures, and memory-generated retrieval clues. RAPTOR organizes recursive summaries into a tree (Sarthi et al., 2024); GraphRAG, LightRAG, and HippoRAG use graph-based corpus representations (Edge et al., 2024; Guo et al., 2024; Gutiérrez et al., 2024); and MemoRAG uses global memory to generate retrieval clues (Qian et al., 2025). LongRefiner similarly exploits hierarchical structure to reduce redundant long-document context (Jin et al., 2025). Dense, sparse, and late-interaction retrievers provide complementary matching signals (Karpukhin et al., 2020; Formal et al., 2021; Santhanam et al., 2022; Thakur et al., 2021). Large-scale approximate-nearest-neighbor systems make billion-point indexing practical, although their recall–latency trade-offs concern candidate access rather than downstream evidence sufficiency (Jayaram Subramanya et al., 2019; Indyk and Xu, 2023). MegaMem combines a distilled retrieval view with a detailed-evidence view and fuses their rankings using reciprocal rank fusion (Cormack et al., 2009). Unlike methods that directly expose summaries or graph nodes to the answer model, MegaMem uses compressed objects only as retrieval keys and resolves them to detailed source evidence before candidate selection and generation.

Query transformation, reranking, and attribution. Query expansion and rewriting reduce vocabulary mismatch in both sparse and dense retrieval (Wang et al., 2023a; Mao et al., 2021; Ma et al., 2023a; Chuang et al., 2023; Liu and Zhang, 2025). Cross-encoders and sequence-to-sequence rerankers further refine broad first-stage candidate pools (Nogueira and Cho, 2019; Nogueira et al., 2020; Ma et al., 2023b). Attribution addresses a separate problem: an answer may be correct while citing unsupported, unnecessary, or incomplete sources (Gao et al., 2023; Min et al., 2023; Es et al., 2024; Qi et al., 2024). MegaMem makes this distinction explicit: retrieval and reranking determine the evidence shown to the answer model, whereas post-answer attribution determines which loaded sources are finally reported.

Heterogeneous and enterprise retrieval. Enterprise corpora combine heterogeneous sources, private terminology, version conflicts, and unanswerable requests. EKRAG evaluates retrieval over corporate reports and product knowledge (Yu et al., 2025); Deep Search covers linked repositories, meetings, messages, and answerability (Choubey et al., 2025); and EnterpriseRAG-Bench scales evaluation to roughly half a million artifacts across nine source types (Sun et al., 2026). We use EnterpriseRAG-Bench as the primary evaluation setting and further test FinanceBench, HotpotQA, LoCoMo, and UltraDomain to assess transfer across financial, multi-hop, conversational, and cross-domain retrieval regimes.

B Full Experimental Protocol

Table 7 summarizes the complete experimental design, including the question split, memory scale, extractor, evidence setting, answer model, and purpose of each analysis. Because these

protocols differ in data partition and system configuration, their absolute values should be interpreted only within the corresponding experiment.

The system is developed on 100 questions and then frozen before evaluation on the disjoint 400-question validation split. The full-benchmark ablation, sampled evidence-efficiency study, mechanism diagnostics, and external transfer experiments are reported separately and are not used as substitutes for the main validation results. EnterpriseRAG-Bench metrics follow the benchmark's canonical gold-answer evaluation protocol. Query expansion recovers 13 of 33 targets missed by the base retriever, while post-answer attribution reduces the average reported source set from 5.00 to 2.31 documents.

Backend and vector storage. We implement the retrieval backend with disk-persistent ChromaDB stores using `PersistentClient`, with a separate persistence directory for each memory scale (Chroma, 2026a). The 60M, 100M, and 250M stores contain 49,528, 82,777, and approximately 190K documents, respectively, with the 250M store occupying approximately 60 GB on disk. Each store contains two independently embedded and queried collections: `raw_chunks`, which stores deterministic chunks of approximately 400 `cl100k_base` tokens, and `distilled_memory`, which stores up to three typed memory entries derived from each chunk. Collections are created with `get_or_create_collection` and reused in read-only mode during inference (Chroma, 2026b). Both collections use `text-embedding-3-small` with 1,536-dimensional vectors (OpenAI, 2024) and ChromaDB's HNSW approximate-nearest-neighbor index with cosine distance. The two collections are queried independently and fused after distilled hits are resolved to their source chunks, implementing the dual-index retrieval component evaluated in our ablation.

C Diagnosing Retrieval Degradation with Gold Evidence

We use a gold-evidence intervention to determine whether performance degradation at larger memory scales is caused primarily by retrieval or by answer generation. For each memory scale, GPT-5.4 answers either from the evidence retrieved by MegaMem or from the benchmark gold evidence. If generation were the main bottleneck, Correctness would decline similarly in both settings. Instead, performance remains nearly stable with gold evidence but drops substantially with retrieved evidence.

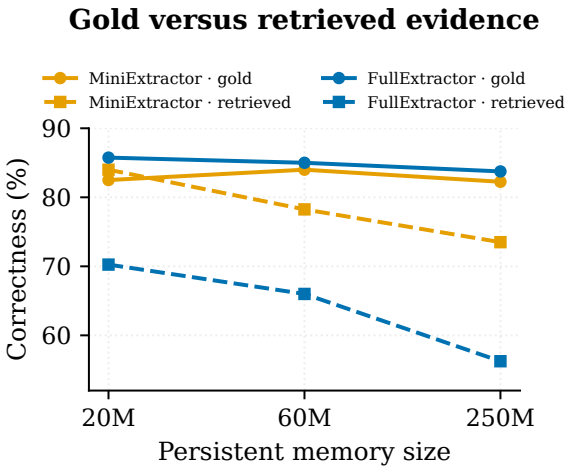

Figure 4: **Separating retrieval failure from generation failure.** Correctness remains stable when GPT-5.4 receives gold evidence, whereas it declines with retrieved evidence as persistent memory grows.

From 20M to 250M tokens, gold-evidence Correctness changes by only 0.30% with MiniExtractor and 2.33% with FullExtractor, compared with 12.50% and 19.93% under retrieved evidence. This contrast shows that most of the observed scaling loss arises from retrieving the right evidence from a larger corpus rather than from the answer model's ability to use that evidence.

D From Compressed Memory to Source-Resolved Dual-View Retrieval

This section summarizes the architecture development that led to the final MegaMem design. We first explored hierarchical and relation-based memory structures, which compress the corpus into coarse summaries or semantic links. Although these representations reduced the search space, they did not reliably recover the exact source evidence required for answering. This limitation motivated the shift to dual-view retrieval, where compact memories support semantic access and detailed chunks preserve source-faithful evidence.

Table 7: **Complete experimental protocol.** The 100-question development split is used only for architecture, prompt, and hyperparameter selection. All remaining analyses preserve their own question sets, memory scales, and evaluation purposes.

Analysis	Questions	Memory scale	Extraction	Evidence setting	Answer model	Purpose
Main comparison	400 validation	10M	FullExtractor	expansion + attribution	GPT-5.4	end-to-end quality
Scaling study	400 validation	20–250M	MiniExtractor	dual-view retrieval	GPT-5.4	scalability
Gold intervention	400 validation	20M, 60M, 250M	both	retrieved / gold	GPT-5.4	failure localization
Component ablation	all 500	10M	FullExtractor	one component removed	GPT-5.4	component contribution
Evidence efficiency	100 per scale	10M, 20M	retained stack	depth 20	GPT-5.4	token–quality trade-off
Mechanism diagnostics	all 500	10M	FullExtractor	varies	GPT-5.4	retrieval and attribution analysis
External transfer	task-specific	task-specific	FullExtractor	task-specific	GPT-5.4	cross-dataset transfer

Table 8: **Correctness with retrieved and gold evidence across memory scales.** Both extractors use the 400-question validation split and GPT-5.4 answering. Change is measured from 20M to 250M tokens.

Extractor	Evidence source	20M	60M	250M	Change (%)
MiniExtractor	Gold	82.50	84.00	82.25	−0.30
MiniExtractor	Retrieved	84.00	78.25	73.50	−12.50
FullExtractor	Gold	85.75	85.00	83.75	−2.33
FullExtractor	Retrieved	70.25	66.00	56.25	−19.93

Table 9 summarize the architecture development process. The first two variants organize extracted concept memories through hierarchical or relational structure, while Cognitive Document Memory combines these forms of concept organization, following prior work on long-term conversational memory (Maharana et al., 2024). Dual-view memory then replaces concept-only retrieval with parallel distilled-memory and detailed-evidence indices. Later variants strengthen memory extraction and source resolution, add cross-encoder reranking (Nogueira and Cho, 2019), and introduce query expansion (Wang et al., 2023a). The final attribution stage identifies the loaded sources supporting the fixed answer (Gao et al., 2023). Because several components change between successive rows, these results describe the architecture development trajectory rather than a controlled ablation.

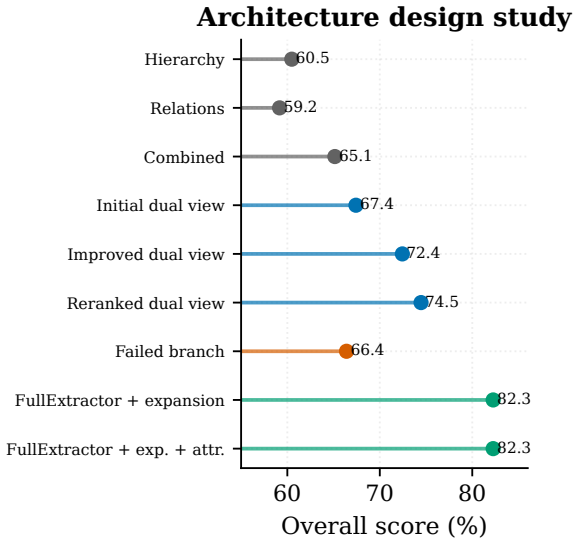

Figure 5: **Performance across successive architecture designs.** Dual-view retrieval provides the main improvement over compressed-only memory, while reranking, query expansion, and attribution further refine evidence selection and source reporting.

Development stages. Hierarchy-only and relation-only memory use compressed representations as the primary access structures. The initial dual-view system introduces separate detailed-evidence and typed-memory indices, while the improved variant refines source resolution and evidence construction. The reranked dual-view system adds cross-encoder ordering, and FullExtractor with query expansion further improves semantic recall. Post-answer attribution is added last and improves source reporting without changing answer quality. Overall, the development path shows that the key architectural change is the transition from compressed-only memory to source-resolved dual-view retrieval.

Table 9: **Architecture development on EnterpriseRAG-Bench.** Each system is selected using the fixed 100-question development split and evaluated on the disjoint 400-question validation split. The rows record successive architecture changes rather than leave-one-component-out effects. InvDocs is bounded in $[0, 1]$, and lower is better.

System	Overall	Document Recall (%)	Correctness (%)	Completeness (%)	InvDocs
Hierarchy-only concept memory	60.48	68.59	73.00	64.19	0.818
Relation-only concept memory	59.16	66.74	72.50	63.02	0.828
Cognitive Document Memory	65.14	70.53	76.75	68.83	0.816
Cognitive Document Memory	66.03	70.53	78.25	69.50	0.816
Initial dual-view memory	67.42	74.80	78.00	71.21	0.804
Improved dual-view memory	72.44	76.03	83.25	76.35	0.802
Reranked dual-view memory	74.46	75.78	84.25	77.58	0.804
Expanded-control branch	66.40	74.69	85.50	69.63	0.806
FullExtractor + Expansion	82.26	81.90	86.50	86.98	0.774
FullExtractor + Expansion + Attribution	82.26	81.90	86.50	86.98	0.760

E Offline Cost, Online Latency, and Evidence-Context Efficiency

This section analyzes the operational cost of MegaMem and the trade-off between evidence-context size and answer quality. Table 10 separates one-time memory construction from online retrieval, answering, and evaluation.

Table 10: **Offline construction, online serving, and evaluation cost.** Build times are reported for 60M and 250M tokens, while serving latency is measured at 60M and 100M.

Regime	Measurement	Value
Offline construction	60M memory	9.3 h
Offline construction	250M memory	approximately 5.6 d
Online retrieval	60M / 100M	0.30 s / 1.10 s
End-to-end answering	60M / 100M	2.5 s / 3.3 s
Evaluation	500 questions / per question	\$3.75 / \$0.0075

Although memory construction becomes more expensive as the corpus grows, it is performed offline. Online retrieval remains below 1.1 seconds at 100M tokens, and end-to-end answering remains below 3.3 seconds, showing that the bounded evidence context keeps inference practical.

We next compare distilled-only evidence, selective detailed evidence used by MegaMem, and full detailed evidence. All settings use the same 100 questions, candidate depth of 20, and GPT-5.4 answer model.

At both memory scales, MegaMem reduces the answer context by approximately 69% relative to full detailed evidence, while sacrificing only 1.5 Correctness points. Distilled-only evidence is shorter but performs substantially worse, confirming that compressed memories are effective retrieval keys but insufficient as generation evidence.

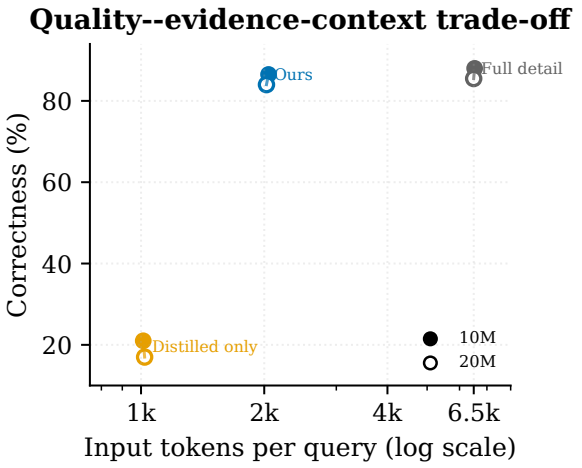

Figure 6: **Accuracy–context trade-off across evidence policies.** MegaMem preserves nearly all of the Correctness of full detailed evidence while using approximately one third of the answer-model input.

Table 12 reports the per-query workload for the full 500-question diagnostic. Questions are short on average, while answer length is more variable; the system retrieves approximately 34 chunks before reranking and evidence packing.

F Cross-Dataset Transfer and Query-Expansion Diagnostics

This section analyzes two complementary questions: how well MegaMem transfers across retrieval regimes, and how query expansion affects recall and abstention.

Cross-dataset transfer. Figure 7 visualizes the results in Table 5. MegaMem performs strongly on FinanceBench and Hot-

Table 11: **Answer quality and context size under different evidence policies.** Token counts are measured per question at 10M and 20M persistent-memory scales.

	10M memory		20M memory	
Evidence policy	Tokens/query	Correctness	Tokens/query	Correctness
Distilled only	1,013	21.00	1,020	17.00
Selective detail (MEGAMEM)	2,049	86.50	2,025	84.00
Detailed evidence only	6,532	88.00	6,497	85.50
Detailed/MEGAMEM token ratio	3.19×		3.21×	

Table 12: **Per-query workload under the 500-question diagnostic protocol.**

Component	Mean	Median	P95
Question input tokens	37	34	63
Answer output tokens	439	221	1,622
Retrieved chunks	34.3	35	–

potQA, where evidence is provided within a question-local corpus, but is less effective on LoCoMo and UltraDomain, which require retrieval over broader conversational or cross-domain stores. This contrast indicates that the current document-oriented dual-view index transfers best when evidence units and corpus structure are well aligned with its memory representation.

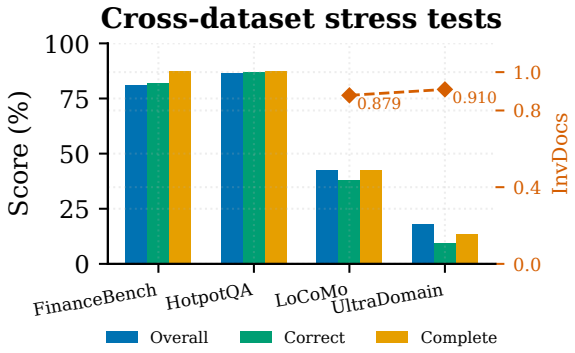

Figure 7: **Transfer performance across evidence regimes.** FullExtractor and GPT-5.4 are evaluated on 150 FinanceBench questions and 200 questions for each remaining benchmark. Document-level metrics are undefined for FinanceBench and HotpotQA.

Retrieval and attribution effects. Query expansion recovers 13 of 33 targets missed by the base retriever, corresponding to a 39.4% recovery rate. Post-answer attribution reduces the average reported source set from 5.00 to 2.31 documents, removing 53.8% of retrieved sources without changing the answer.

Table 13: **Mechanism-level retrieval and attribution diagnostics.** Both analyses use all 500 EnterpriseRAG-Bench questions at 10M with FullExtractor.

Diagnostic	Result
Query-expansion target recovery	13/33 (39.4%)
Evidence pool → reported documents	5.00 → 2.31 (53.8% removed)

Query expansion and answerability. Table 14 shows that query expansion improves Correctness across all five answerable question categories, with relative gains ranging from 20.8% to 100.0%. The largest gains appear on project-related and high-level questions, where terminology mismatch and indirect references make direct retrieval difficult.

Table 14: **Correctness by question type before and after query expansion.** All 500 EnterpriseRAG-Bench questions are evaluated at 10M with FullExtractor.

Question type	Pre-exp. (%)	Unconditional (%)	Selective oracle (%)
Project-related	53.10	96.90	96.90
High-level	37.50	75.00	75.00
Constrained	66.70	87.50	87.50
Conflict	81.20	100.00	100.00
Semantic	72.00	87.00	87.00
Info not found	100.00	68.80	100.00

The main trade-off appears on information-not-found questions. Unconditional expansion answers five of the 16 unanswerable cases, reducing Correctness from 100.0% to 68.8%. A selective oracle preserves all gains on answerable questions while restoring information-not-found Correctness to 100.0%. This result shows that query expansion and reliable abstention are compatible, but require an answerability-aware gate that decides when expansion should be applied.

G Memory Construction, Inference Order, and Reproducibility

Dual-view memory construction. Algorithm 2 gives the complete construction procedure. Each distilled memory is stored with the immutable identifier of the detailed source chunk from which it is derived, ensuring that both indices refer to the same underlying evidence.

Algorithm 2 Construct source-resolved dual-view memory

Require: documents $\mathcal{D}$, memory extractor $\mathcal{E}_m$, embedding function ϕ
for all detailed source chunks x_i in $\mathcal{D}$ **do**
  add $(i, x_i, \phi(x_i))$ to the detailed index
  for all distilled memory $m \in \mathcal{E}_m(x_i)$ **do**
    add $(m, \phi(m), \pi(m) = i)$ to the distilled index
  end for
end for
return detailed index, distilled index, and source map π

Inference and attribution. Figure 2 summarizes the online pipeline. Original and transformed queries first retrieve candidates from both indices. Distilled hits are then resolved to detailed source chunks before fusion, reranking, and evidence packing. The answer is generated from the resulting bounded evidence context, after which attribution selects only the loaded sources that support the fixed answer.

Algorithm 3 Generate an answer and attribute supporting sources

Require: question q, bounded detailed evidence E
$a \leftarrow \textsc{EvidenceOnlyAnswer}(q, E)$
$Z' \leftarrow \textsc{Attribute}(a, E)$
$Z \leftarrow Z' \cap \mathrm{docs}(E)$
return answer a and supporting sources Z

Reproducibility settings. All generative calls use temperature 0, and data splitting, sampling, and tie-breaking use seed 42. Each request has a 120-second timeout and at most two retries with provider backoff. The evidence context is capped at 4,096 tokens, the generated answer at 800 tokens, and the reranked candidate set at five unique documents. RRF uses $\gamma = 60$, while the context-packing study fixes the first-stage candidate depth at 20.

Models and failure handling. Atomic memory extraction uses `gpt-5.4-mini` (OpenAI, 2026b); higher-level abstraction, query transformation, answering, post-answer attribution, and EnterpriseRAG evaluation use `gpt-5.4` (OpenAI, 2026a); and all indices use `text-embedding-3-small` (OpenAI, 2024). Extraction, query transformation, and attribution return schema-validated JSON. After all retries fail, extraction returns no memories, query transformation falls back to the original question, attribution returns an empty set, and answering returns the predefined abstention response. Parsed citations are finally intersected with the document identifiers included in the evidence context, preventing unsupported sources from being reported.

H Visualization of Main Results and Diagnostics

This section visualizes the main system comparison, component ablation, query-expansion behavior, and token usage. Exact values are reported in the corresponding tables.

Figure 8 summarizes end-to-end performance at the 10M memory scale. MegaMem achieves the strongest overall result among direct retrieval baselines and compressed-memory variants, while also maintaining competitive document recall and a lower invalid-document ratio.

Figure 9 shows how each component contributes to the full system. Removing any major component reduces Overall, confirming that dual-view memory, reranking, query expansion, and attribution support complementary parts of the final pipeline.

Figure 10 examines how query expansion affects different question types. Expansion improves performance across answerable categories, but information-not-found questions require an answerability-aware gate to prevent expanded queries from retrieving plausible but unsupported evidence.

I Why Dual-View Retrieval Helps and Where Scaling Fails

This section provides a simple interpretation of the two main empirical findings: dual-view retrieval improves evidence recall, while performance degradation at larger memory scales is driven mainly by retrieval.

Let A denote the event that relevant evidence is retrieved from the detailed index, M the

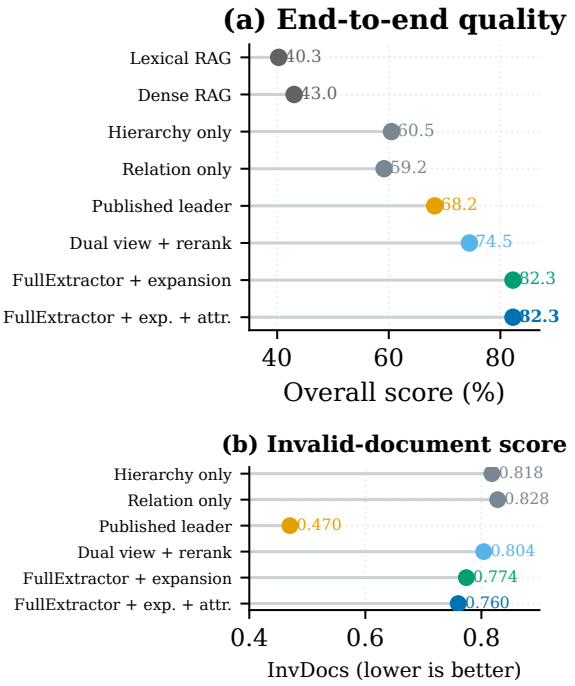

Figure 8: **End-to-end performance at the 10M memory scale.** The figure summarizes the 400-question validation results in Table 3, comparing direct retrieval baselines, compressed-memory variants, and MegaMem. InvDocs is in $[0, 1]$, and lower is better.

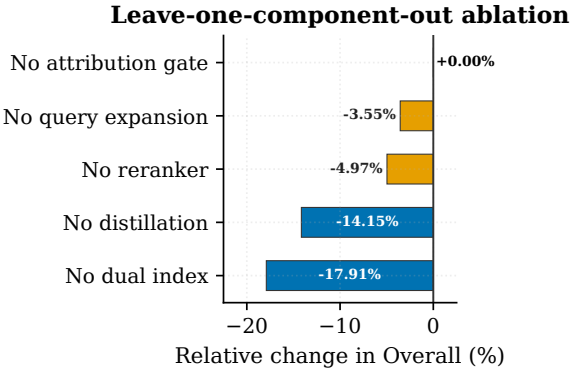

Figure 9: **Contribution of individual MegaMem components.** The figure shows the relative change in Overall after removing each component from the full system under the 500-question ablation protocol.

event that it is retrieved from the distilled index, and G the event that a distilled hit is correctly resolved to its detailed source chunk. The recall of dual-view retrieval is

$$\begin{aligned} R_{\text{dual}} &= \Pr\left(A \cup (M \cap G)\right) \\ &= R_A + \Pr(M \cap G \cap A^c), \end{aligned} \tag{8}$$

where the second term captures evidence recovered by the distilled route but missed by the

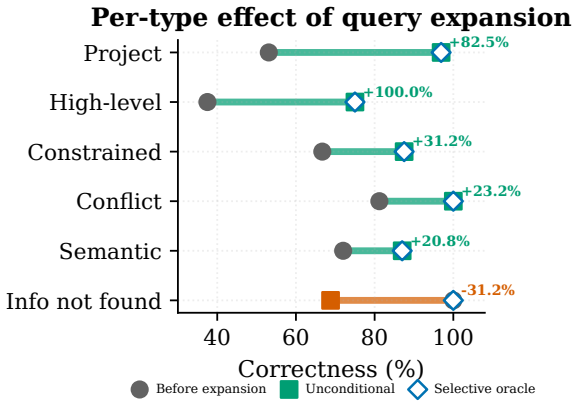

Figure 10: **Effect of query expansion across question types.** Query expansion improves all answerable categories, while an answerability-aware gate is required to preserve performance on information-not-found questions.

detailed route. This term explains the complementarity observed in the ablation study.

To interpret scaling behavior, let S_N denote the event that sufficient evidence is selected from a corpus of size N. Answer Correctness can be written as

$$\Pr(Y = 1 \mid N) = \Pr(S_N)\Pr(Y = 1 \mid S_N) \tag{9}$$

$$+ \Pr(\neg S_N)\Pr(Y = 1 \mid \neg S_N). \tag{10}$$

As memory grows, the main changing term is $\Pr(S_N)$, since relevant evidence must be identified among more distractors. The gold-evidence experiment approximately controls for evidence sufficiency and shows that Correctness remains stable once the required evidence is provided. This supports the conclusion that the observed scaling loss is primarily caused by retrieval rather than answer generation.

J Prompt Templates

This section provides the prompt templates used for memory construction, query transformation, candidate selection, answer generation, attribution, and evaluation. The prompts are reproduced in their original form for completeness and reproducibility.

Atomic typed-memory extraction

System: Extract at most three atomic, retrieval-friendly memories entailed by the source chunk. Each memory must have a type from {fact, procedure, definition, requirement, decision}, a short retrieval key, a concise value, and no outside knowledge. Skip filler. Return `{"memories":[...]}`; return an empty list when the chunk has no useful content.
User: Document chunk, preceded by its section path and immutable chunk identifier.

High-level memory abstraction

System: Summarize the supplied typed memories into a compact search key for their shared topic. Preserve named entities, constraints, dates, exceptions, and conflicts. Do not create a fact absent from the children. Return JSON containing `search_key`, `summary`, and the unchanged list of child source identifiers.
User: A bounded group of typed memories with their immutable document mappings.

Query transformation

System: Preserve the user's information need and named constraints. Produce a canonical rewrite and terminology-diverse alternatives; do not invent an answer, entity, or date. Return a JSON list. If no safe reformulation exists, return only the original query.
User: The original question.

Cross-route candidate selection

System: Select only candidate identifiers that could answer the question under its stated entities, time range, and constraints. A typed-memory match is a search clue, not evidence; retain its linked detailed-evidence identifier. Return ranked identifiers and short selection reasons in JSON. Never invent or rewrite an identifier.
User: Question, fused detailed and typed candidates, route ranks, and source metadata.

Evidence-only answer

System: Answer using only the supplied document evidence. Preserve qualifiers and conflicts. If the evidence is insufficient, reply exactly: `I don't have enough information to answer.`
User: The question followed by evidence cards labeled with chunk identifier, document identifier, and section path.

Post-answer attribution

System: The answer is fixed. Return only identifiers of supplied evidence cards that directly support a claim in that answer. Do not add a source, revise the answer, or cite topically related but unused evidence. Return `{"document_ids":[...]}`.
User: Fixed answer plus the same evidence cards shown to the answer model.

EnterpriseRAG correctness evaluation

System: Compare the candidate answer with the benchmark gold answer. Judge whether the candidate preserves the required conclusion, scope, conditions, and conflicts. Do not reward unsupported details. Return schema-constrained JSON with `correct`, `validated_facts`, and `missing_facts`; do not expose or use any field outside the canonical gold-answer protocol.
User: Question, candidate answer, and canonical gold answer.