LMI: LATENT MEMORY INTERACTION FOR EFFICIENT AND FAITHFUL LONG-CONTEXT COMPRESSION IN LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) have demonstrated strong long-context understanding, yet scaling context length remains challenging due to the quadratic complexity of self-attention and the linear growth of the key-value (KV) cache. Existing compression approaches—token pruning, pooling, and hierarchical summarization—often suffer from irreversible information loss and limited global interaction among compressed representations. We propose **LMI** (Latent Memory Interaction), a window-parallel framework that partitions long sequences into context windows and introduces learnable latent tokens that aggregate and exchange information across windows. Unlike methods that independently compress each segment, LMI dynamically refines latent representations through cross-window interaction, preserving local detail and global semantic dependencies while maintaining near-linear complexity and compatibility with existing LLM architectures. Extensive experiments on long-context benchmarks demonstrate that LMI achieves competitive performance against existing token-reduction and compression methods; on a 128K-token, 32-request serving workload, LMI reduces end-to-end inference time by 8.7× relative to full-context inference, and in a real SWE-bench agent pipeline it raises task success by 7.84 points while cutting wall-clock time by 1.9×.

1 INTRODUCTION

Large language models (LLMs) have demonstrated remarkable capabilities across a wide range of tasks, including complex reasoning, document understanding, and multi-turn dialogue Xu et al. (2026a); Yang et al. (2025); Guo et al. (2025); Comanici et al. (2025). As these capabilities continue to evolve, enabling LLMs to effectively process increasingly longer contexts has become a critical research direction. Recent models have extended their context windows from thousands to hundreds of thousands of tokens, allowing them to incorporate richer information from long context Xu et al. (2025); Zhao et al. (2024); Gao et al. (2026); Bai et al. (2024a); Wan et al. (2025); Shen et al. (2025b). However, the practical deployment of long-context LLMs remains fundamentally constrained by the computational and memory overhead introduced by the self-attention mechanism.

A natural solution is context compression, which identifies and preserves the most informative parts of a long input while removing redundancy Joo et al. (2026); Jiang et al. (2024); Pan et al. (2024); Liu & Yu (2025); Louis et al. (2025); Xu et al. (2026b); Dai et al. (2025); Li et al. (2025b); Cui et al. (2025). Existing approaches fall into three categories—token pruning, token merging, and LLM-based summarization—but they typically treat tokens independently, introduce irreversible information loss, or incur additional inference cost while failing to preserve the fine-grained evidence required by knowledge-intensive tasks.

Beyond information loss, we identify another fundamental limitation of existing compression paradigms: the lack of global interaction during compression. Most existing methods independently process different segments of a long document, assuming that the importance of each token can be determined locally Li et al. (2025a); Yu et al. (2025); Zhang et al. (2025); Ge et al. (2025); Shen et al. (2025a). However, in many long-context scenarios, such as multi-hop question answering and needle-in-a-haystack retrieval, critical information may be distributed across distant regions

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

of the document. Compressing each segment independently prevents different context regions from exchanging information, leading to incomplete representations and degraded retrieval accuracy.

To address these challenges, we propose **LMI** (**L**atent **M**emory **I**nteraction), a novel learned compression framework for scalable long-context LLM inference. The key idea of LMI is to introduce a set of learnable latent memory tokens into each context window, allowing the model to aggregate and exchange contextual information through attention-based interaction. Specifically, a long document is first partitioned into multiple chunks, and latent tokens are appended to each chunk. Through self-attention computation, these latent tokens absorb the semantic information contained within their corresponding windows. Unlike conventional window-wise compression methods, LMI enables latent memories from different windows to interact with each other, forming a global information exchange pathway across the entire document. This design allows LMI to process arbitrarily long contexts with near-linear complexity while preserving long-range dependencies.

Furthermore, we observe that effective context compression requires more than simply assigning importance scores to individual tokens. Direct token-level pruning often introduces semantic fragmentation, where critical sentence-level information becomes incomplete after compression Joo et al. (2026); Pan et al. (2024); Liu & Yu (2025). To address this issue, LMI introduces a semantic-aware restructuring strategy that aggregates token-level predictions into sentence-level importance estimation and performs structured retention.

To train an effective compression model, we further develop a large-scale data generation pipeline based on a data flywheel strategy. Instead of relying on heuristic token-level labels, we construct high-quality supervision signals by measuring the influence of each sentence on the model output distribution through perplexity-based perturbation analysis. Based on these automatically generated datasets, we develop both a 7B-scale SFT compression model and a distilled 0.5B lightweight variant. Extensive evaluations on diverse long-context benchmarks demonstrate that LMI strikes a more favorable compression-performance trade-off compared to existing context compression approaches, reducing end-to-end serving latency by 8.7$\times$ on 128K-token workloads.

Our contributions are summarized as follows:

- We introduce LMI, a novel latent memory interaction framework for long-context compression, where learnable latent tokens enable cross-window information exchange and efficient global context aggregation with near-linear complexity.
- We propose a semantic-aware compression strategy, which converts token-level importance estimation into structured sentence-level retention, alleviating semantic fragmentation caused by naive token pruning.
- We construct a data flywheel training pipeline, generating high-quality compression supervision through perplexity-based importance estimation.

2 Related Work

2.1 Context Compression for Large Language Models

Existing approaches can be broadly categorized into token-level compression, representation-based compression, and generation-based compression.

Token-level compression methods identify and remove redundant tokens while preserving task-relevant information. LLMLingua Jiang et al. (2024) and LLMLingua-2 Pan et al. (2024) estimate token importance to prune less informative tokens; PISCO Louis et al. (2025) targets retrieval-augmented generation, and Mustafar Joo et al. (2026) uses dynamic token selection to reduce inference memory.

QwenLong-CPRS Shen et al. (2025a) is the most closely related system to LMI: it likewise adopts window-parallel processing, a hybrid causal/bidirectional compressor architecture, and token-level content selection. LMI differs from CPRS in two respects: (i) learnable latent memory tokens that periodically aggregate and broadcast information across windows, enabling distant context regions to communicate during compression, and (ii) aggregation of token-level predictions into budget-constrained sentence-level retention. These two components are the focus of this work.

Representation-level methods learn compact continuous representations. QuickMerge++ Liu & Yu (2025), K-token merging Xu et al. (2026b), and embedding-based memory compression Dai et al. (2025) aggregate token representations into fewer latent states, but the compression is often irreversible and may discard fine-grained information needed for downstream reasoning.

Generation-based methods use auxiliary LLMs to summarize or rewrite long contexts. 500xCompressor Li et al. (2025b) and Less Is More Cui et al. (2025) generate compact summaries optimized for downstream tasks, but incur extra generation cost and may fail to preserve exact evidence for knowledge-intensive tasks.

2.2 Latent Representation and Memory-Augmented Modeling

Prior works such as Perceiver Jaegle et al. (2021) and related latent bottleneck architectures Hawthorne et al. (2022); Hiller et al. (2024) demonstrate that a small number of latent variables can efficiently capture information from high-dimensional inputs through cross-attention mechanisms. Inspired by this idea, recent long-context compression methods have investigated latent representations as compact memory carriers.

Infini-attention Munkhdalai et al. (2024) continuously aggregates history into compact memory states; MemoryLLM Wang et al. (2024) maintains self-updatable latent memory slots; Activation Beacon Zhang et al. (2025) compresses history via additional activation tokens; BRIEF Li et al. (2025a) bridges retrieval and inference in multi-hop reasoning; and Context Compressor Dai et al. (2025) learns dense memory representations. These approaches demonstrate the potential of learned memory representations for efficient long-context processing.

LMI differs from these methods by introducing a window-parallel latent memory interaction mechanism, where latent tokens from different context regions communicate with each other to construct a globally consistent compressed representation.

3 Method

3.1 Overview

Given a long context sequence ($X = x_1, \ldots, x_N$), our goal is to identify and retain the most informative content while substantially reducing the sequence length for downstream inference. As illustrated in Figure 1, we introduce LMI (Latent Memory Interaction), a learned context compression framework that combines global information interaction with semantic-aware structured retention.

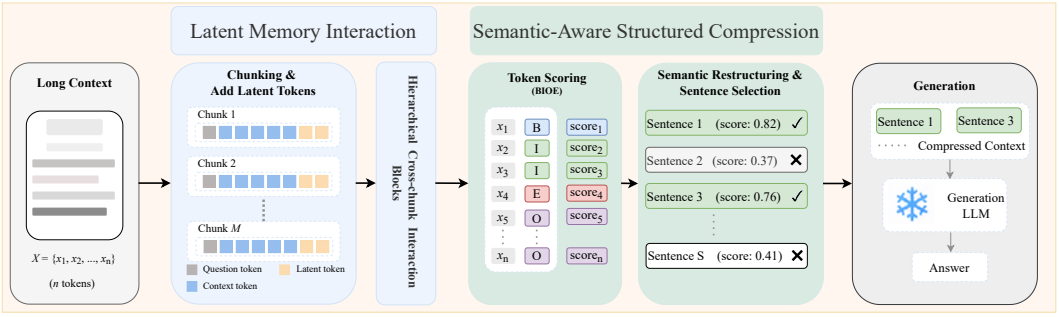

Figure 1: Overview of LMI. Given a long context, LMI first partitions the sequence into multiple chunks and introduces learnable latent tokens. The latent tokens enable cross-chunk interaction via hierarchical attention. Then, a BIOE token classification is performed, followed by semantic-aware structured compression. The compressed context is finally fed into the LLM for generation.

3.2 Latent Memory Interaction

Processing arbitrarily long contexts with full self-attention is computationally prohibitive because its complexity grows quadratically with the sequence length. A straightforward solution is to divide

a long context into independent chunks and process them separately. However, independent processing of individual chunks hinders cross-region information interaction. To address this issue, we introduce a set of learnable latent memory tokens into each chunk. Specifically, the input sequence is partitioned into M chunks, where each chunk X_m contains at most K input tokens. We append R learnable latent tokens $Z_m = z_{m,1}, \ldots, z_{m,R}$ to the end of each chunk, yielding $[X_m; Z_m]$.

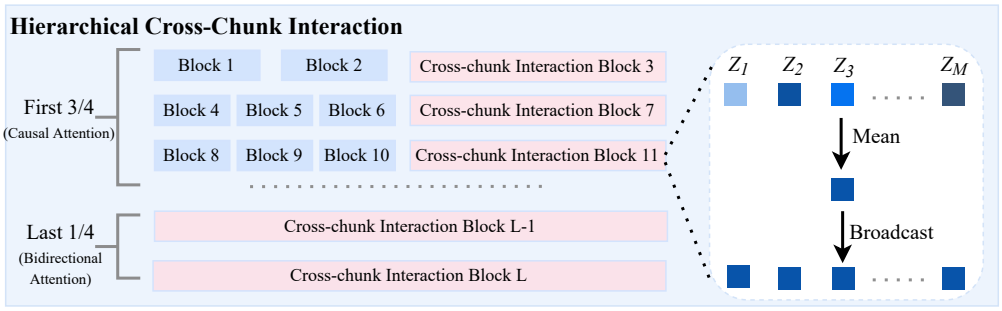

Figure 2: Hierarchical cross-chunk interaction. In the first three quarters of the layers, latent tokens (highlighted) from different chunks are aggregated via an element-wise mean and broadcast back to every chunk at selected interaction layers (*e.g.*, layers 3, 7, 11); the final quarter of layers employs bidirectional attention among latent memories for global refinement.

As illustrated in Figure 2, during self-attention, latent tokens attend to the contextual representations preceding them and progressively aggregate information from their corresponding chunks. This design provides an information bottleneck through which a potentially long chunk can be summarized into a small number of latent representations. Although latent tokens can summarize their local chunks, independently computed latent memories still cannot capture dependencies between distant context regions. We therefore introduce cross-chunk latent memory interaction inside the transformer. A key challenge is that the compressor inherits the causal attention structure of the underlying LLM. Directly replacing causal attention with bidirectional attention throughout the network would alter the model's inductive bias and substantially increase the architectural deviation from the pretrained model. Instead, we employ a hierarchical attention schedule.

Specifically, the first three quarters of the transformer blocks retain the original causal attention mechanism, while the final quarter of the blocks employ bidirectional attention. Within the first stage, latent memories from different chunks periodically interact at selected transformer layers. For example, given the layer indices l, cross-chunk interaction is performed at $l \in [3, 7, 11, \ldots]$. At these interaction layers, only the latent memory tokens at the end of each chunk are extracted.

Let $\left[Z_1^{(l)}, Z_2^{(l)}, \ldots, Z_M^{(l)}\right]$ denote the collection of latent memories across all chunks. We compute their element-wise mean:

$$\bar{Z}^{(l)} = \frac{1}{M} \sum_{m=1}^{M} Z_m^{(l)}, \tag{1}$$

and broadcast the resulting latent memory back to every chunk:

$$Z_m^{(l)} \leftarrow \bar{Z}^{(l)}, \qquad m = 1, \ldots, M. \tag{2}$$

In other words, the latent tokens act as a compact communication channel between different chunks. The interaction operates exclusively on R latent tokens rather than the original K input tokens.

In the final quarter of the transformer blocks, we enable continuous bidirectional interaction among latent memories. This allows the compressed representations to undergo further global refinement after the periodic information exchange in the earlier layers. The resulting architecture therefore combines local causal encoding, sparse global communication, and late-stage bidirectional refinement. Detailed analysis of computational complexity and inference efficiency is provided in Appendix A.1.

3.3 SEMANTIC-AWARE STRUCTURED COMPRESSION

The latent interaction module produces contextualized token representations that are subsequently used to estimate the importance of individual tokens. Directly pruning tokens according to token-level scores can result in fragmented sentences and incomplete semantic units. To mitigate this problem, LMI separates token-level importance estimation from structured content retention.

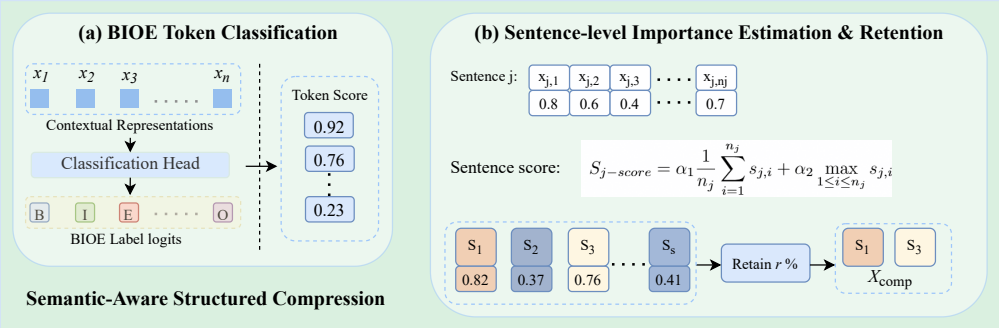

Figure 3: Semantic-aware structured compression. The compressor predicts four-way BIOE labels for each token; token-level scores $s_i = p_i^I - p_i^O$ are aggregated into sentence-level importance via a weighted mean–max combination, and complete sentences are retained under a predefined compression budget, preserving grammatical structure and avoiding fragmented outputs.

As illustrated in Figure 3, for every input token x_i, the compressor predicts a four-way label corresponding to the BIOE scheme, where B, I, O, and E denote the beginning, interior, outside, and ending positions of a retained semantic unit, respectively. Let h_i denote the contextual representation of token x_i. A lightweight classification head produces

$$p_i = \mathrm{Softmax}(W h_i + b), \tag{3}$$

where $p_i \in \mathbb{R}^{B,I,O,E}$ represents the probability distribution over the four BIOE labels. We derive a scalar importance score for token x_i by contrasting the probability of the token being inside a retained semantic unit with that of being outside it:

$$s_i = p_i^I - p_i^O, \tag{4}$$

the B and E probabilities are primarily used to capture the structural boundaries of retained semantic units and are therefore not directly involved in the scalar importance score.

We next aggregate token-level importance scores into sentence-level scores. Let a sentence S_j contain n_j tokens with importance scores $[s_{j,1}, s_{j,2}, \ldots, s_{j,n_j}]$. We define the sentence importance as a weighted combination of the mean and maximum token importance:

$$S_{j-score} = \alpha_1 \frac{1}{n_j} \sum_{i=1}^{n_j} s_{j,i} + \alpha_2 \max_{1 \le i \le n_j} s_{j,i}, \tag{5}$$

where α_1 and α_2 are predefined hyperparameters.

The mean component captures the overall semantic relevance, while the maximum component captures the presence of highly informative local evidence. This combination prevents two undesirable behaviors. First, a sentence containing many moderately important tokens can be recognized as globally relevant even when no individual token dominates. Second, a sentence containing a small but critical piece of evidence can still be retained through its high maximum score.

Based on the sentence-level scores, we rank sentences and retain the highest-scoring semantic units under a predefined compression budget. The retained sentences are concatenated to form the final compressed context:

$$S_{j_1} \oplus S_{j_2} \oplus \cdots \oplus S_{j_r}, \tag{6}$$

where r is determined by the target compression ratio.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

We train the model on both QA and text summarization data with a standard cross-entropy loss over the four-way BIOE labels. For the QA setting, we construct a training corpus consisting of 25K instances aggregated from public benchmarks including SQuAD Rajpurkar et al. (2016), HotpotQA Yang et al. (2018), Musique Trivedi et al. (2022), and 2WikiMultiHop Ho et al. (2020). However, existing document–summary datasets typically provide only document-level supervision in the form of a source document paired with a ground-truth summary, without explicitly identifying the sentences that contribute to the target summary. To bridge this gap, we introduce a Perplexity-Guided Key Sentence Mining procedure that automatically derives sentence-level relevance signals from document–summary pairs. The detailed algorithmic procedure is provided in Appendix A.2. Using this procedure, we automatically generate approximately 23k training samples and 0.5k test samples from publicly available datasets including CASSum fpg2012 (2024), DISC-Law-SFT Yue et al. (2023), and WikiSum Liu et al. (2018). Both compressors are initialized from Qwen2 base backbones—the 7B model from Qwen2-7B and the 0.5B model from Qwen2-0.5B—so that the 7B variant matches the backbone of the CPRS baseline. We first perform standard supervised fine-tuning (SFT) on the 7B-parameter model and subsequently train its 0.5B-parameter counterpart through feature distillation. Key parameters involved in the training and distillation processes are detailed in Appendix A.3.

We evaluate our approach across three task scenarios: multi-turn dialogue, needle-in-haystack, and long-context QA. We benchmark against four competitive baselines: RAG Li et al. (2023), SYNAPSE Jiang et al. (2026), Provence Chirkova et al. (2025), and CPRS Shen et al. (2025a), where the CPRS compressor is a 7B-parameter model sharing the same Qwen2 backbone and scale as LMI-7B. The RAG baseline employs the gte-large-en-v1.5 model as its retrieval backbone.

4.2 MAIN RESULTS

4.2.1 NEEDLE-IN-A-HAYSTACK

We construct Needle-in-a-Haystack evaluation sets of varying difficulty from publicly available corpora Li et al. (2024), including two moderate-difficulty settings, *Needle5_in_Haystack_8K* and *Needle5_in_Haystack_32K*, and a more challenging setting, *Needle5_in_Haystack_32K_atc*. For the moderate-difficulty settings, we sample needles from the Multi-Needle Reasoning corpus. However, we find that multiple needles in this corpus often exhibit strong semantic associations with the question.

To better assess the latter capability, we introduce a more challenging evaluation setting inspired by the Ancestral Trace Challenge (ATC). Specifically, we use randomly sampled names and predefined relational templates to construct multi-hop chains of person-to-person relations. In contrast to the moderate-difficulty settings, the relevant evidence in these instances is distributed across a chain of relations and is not necessarily directly aligned with the query. Successfully solving these instances therefore requires retrieving and jointly integrating multiple interdependent pieces of evidence, providing a more stringent test of latent-token interaction. The results are reported in Table 1.

LMI consistently achieves a favorable trade-off between retrieval accuracy and the amount of retrieved information. On the two moderate-difficulty benchmarks, LMI-7B achieves the highest recall of 92.09% and 90.77%, respectively, while retrieving only 4.88 and 5.53 sentences on average. Compared with CPRS, LMI-7B improves recall by 1.39% and 1.47%, while reducing the number of retrieved sentences by 31.2% and 52.7%, respectively. This indicates that LMI can identify relevant evidence more selectively, avoiding the need to retrieve a large number of potentially redundant sentences. In particular, the advantage of LMI becomes more pronounced in the challenging setting of *Needle5_in_Haystack_32K_atc*. The consistent gains in both recall and retrieval compactness suggest that LMI is particularly effective when the relevant evidence is distributed across multiple relational hops and cannot be identified solely through direct semantic matching. Some visualization results can be found in Appendix A.4.

The distilled 0.5B model also exhibits strong retrieval efficiency. On the 8K and 32K settings, LMI-Distill-0.5B retrieves substantially fewer sentences than the competing baselines while maintaining

Table 1: Comparison of retrieval performance on Needle-in-a-Haystack benchmarks. Recall denotes the percentage of successfully retrieved needles, while Retrieved Sentence Num denotes the average number of retrieved sentences. Qwen2-distill-0.5B refers to the distilled model obtained by directly performing feature distillation from CPRS-7B with Qwen2-base as the student model.

Benchmark	Model	Recall (%) ↑	Retrieved Sentence Num ↓
Needle5_in_Haystack_8K	RAG	85.67	7.00
	Provence	87.25	6.54
	CPRS-7B (chunk=2k)	90.70	7.09
	LMI-Distill-0.5B (chunk=2k)	90.23	5.57
	LMI-7B (chunk=2k)	**92.09**	**4.88**
Needle5_in_Haystack_32K	RAG	82.42	8.00
	Provence	88.47	7.91
	CPRS-7B (chunk=8k)	89.30	11.70
	LMI-Distill-0.5B (chunk=8k)	89.30	7.64
	LMI-7B (chunk=8k)	**90.77**	**5.53**
Needle5_in_Haystack_32K_atc	RAG	49.60	8.00
	Provence	52.13	8.82
	CPRS-7B (chunk=8k)	36.40	10.07
	Qwen2-distill-0.5b (chunk=8k)	31.52	17.92
	LMI-Distill-0.5B (chunk=8k)	60.60	7.68
	LMI-7B (chunk=8k)	**70.80**	**5.45**

Table 2: Comparison of retrieval performance on the LoCoMo benchmark.

Model	RAG	SYNAPSE	Provence	CPRS-7B	LMI-Distill-0.5B	**LMI-7B**
Recall (%) ↑	74.32	80.40	90.24	96.02	95.61	**97.05**

recall comparable to CPRS. Although the distilled model incurs a moderate performance gap relative to LMI-7B, its results demonstrate that a substantial portion of the retrieval capability can be transferred to a much smaller model through feature distillation.

To assess whether the observed gains generalize beyond synthetic Needle-in-a-Haystack settings, we further evaluate LMI on LoCoMo Maharana et al. (2024), a long-conversation benchmark that requires retrieving relevant evidence from extensive multi-turn conversational histories. This task can be viewed as a real-world analogue of the needle-in-a-haystack setting. As shown in Table 2, LMI-7B achieves the best performance with a recall of 97.05%. This consistent improvement on naturally occurring long-conversation data demonstrates that the benefits of latent-token interaction extend beyond controlled synthetic retrieval tasks and generalize to complex conversational contexts.

QwenLong-CPRS Shen et al. (2025a) serves as our closest system-level baseline, given its shared window-parallel hybrid architecture with LMI. Full performance comparisons across all settings are deferred to Appendix A.5.

4.2.2 Question Answering

We further evaluate LMI on long-context question answering using LongBench-v1 Bai et al. (2024b). Specifically, we select instances whose input length exceeds 8K tokens to focus on challenging long-context scenarios. The compression ratio is defined as the ratio between the length of the compressed text and that of the original context. For fairness of the comparison, the compression ratios of all compression methods are roughly aligned with those of LMI-7B. As shown in Table 3, LMI consistently preserves strong downstream QA performance while substantially reducing the amount of context passed to the generation model.

When using Qwen2.5-7B-Instruct as the generation model, LMI-7B achieves an average score of 29.95, matching the uncompressed setting (29.91) and surpassing CPRS (28.07) and Provence (23.12) by 1.88 and 6.83 points; the distilled 0.5B model reaches a comparable 29.36 at a similar compression ratio, with a consistent trend under Qwen2.5-14B-Instruct. LMI thus removes

Table 3: Long-context question answering performance on LongBench-v1 using samples with input length exceeding 8K tokens. The compression ratio is defined as the ratio between the compressed-context length and the original context length. AVG denotes the weighted average over all subtasks.

Generation Model	Compression Model	NQ	DU	VC	Musi	QM	PC	TQ	AVG
Qwen2.5-7B-Instruct	Full Context	25.62	29.30	16.32	29.87	23.55	6.49	87.39	29.91
	RAG	12.15	20.64	9.81	11.14	20.15	1.57	81.15	20.47
	Compression Ratio	14.51%	16.13%	15.62%	9.23%	28.42%	7.10%	10.11%	–
	Provence	16.27	22.54	11.52	17.91	21.25	1.78	82.34	23.12
	Compression Ratio	14.35%	16.25%	15.33%	9.14%	27.95%	7.21%	10.42%	–
	CPRS-7B	24.18	23.56	13.36	34.72	21.80	2.24	83.90	28.07
	Compression Ratio	14.54%	16.17%	15.47%	9.19%	28.20%	7.32%	10.82%	–
	LMI-Distill-0.5B	25.22	27.90	14.23	31.26	23.49	4.86	86.86	29.36
	Compression Ratio	14.41%	15.97%	15.32%	8.98%	27.85%	7.30%	10.75%	–
	LMI-7B	26.30	26.73	14.07	38.95	23.13	2.16	84.65	**29.95**
	Compression Ratio	14.27%	16.44%	16.43%	8.80%	28.19%	6.84%	6.00%	–
Qwen2.5-14B-Instruct	Full Context	27.32	31.20	17.51	32.42	24.52	7.56	88.25	31.44
	RAG	14.25	21.58	11.23	15.18	21.03	1.88	83.52	22.25
	Provence	17.23	24.77	12.75	22.16	22.42	2.65	85.18	25.09
	CPRS-7B	26.70	24.35	12.80	40.68	22.12	2.92	85.83	29.99
	LMI-Distill-0.5B	26.82	26.40	13.43	33.15	23.13	3.26	86.88	29.38
	LMI-7B	27.69	25.54	14.24	43.80	22.99	2.63	86.58	**31.17**

Table 4: End-to-end serving efficiency on 128K-token inputs with 32 concurrent requests and a fixed output length of 10K tokens. Compression Ratio denotes the ratio of the compressed-context length to the original context length. Compression and Generation report the per-task time spent in the compressor and the generation model, respectively.

Method	Total Time (s) ↓	Avg. Task (s) ↓	Comp. (s) ↓	Gen. (s) ↓	Ratio ↓
Full Context (MiniMax-M2.7)	7850.6	1153.2	–	1153.2	–
LMI-7B → MiniMax-M2.7	**902.1**	**137.45**	41.95	**95.5**	10:1

redundant context without sacrificing the information needed for downstream question answering, preferentially preserving content that aids reasoning rather than merely shortening the context.

4.3 Efficiency Analysis

To quantify the practical efficiency gains, we measure end-to-end serving latency on 128K-token inputs with 32 concurrent requests and a fixed 10K-token output length, comparing (i) **Full Context**, where MiniMax-M2.7 directly consumes the full context, and (ii) **LMI**, where the context is first compressed by LMI-7B. Both configurations are served with vLLM on eight Huawei Ascend 910 NPUs.

As shown in Table 4, LMI reduces the total wall-clock time from 7850.6 s to 902.1 s (an **8.7**× speedup) and the average per-task latency from 1153.2 s to 137.45 s (**8.4**×); the one-time compression cost (41.95 s, 30.5% of per-task latency) is more than offset by the **12.1**× reduction in generation time (1153.2 s → 95.5 s), which benefits from a roughly 10× shorter prefill and a proportionally smaller KV cache (Appendix A.1).

4.4 End-to-End Agent Evaluation

To validate LMI beyond retrieval and QA proxies, we embed it in a real SWE-bench Jimenez et al. (2024) agent pipeline: for each issue, BM25 Robertson & Zaragoza (2009)-retrieved code fragments are concatenated into a context exceeding 40K tokens and fed to the OpenCode Contributors (2024) agent driven by GLM-5.1 Team & AI (2026) (W4A8) as a single first-round prompt. We compare **OpenCode** (full BM25 context) against **LMI + OpenCode** (LMI-7B first compresses the context at 3:1), both on 16 Ascend 910 NPUs (TP=16, DP=1).

As shown in Table 5, LMI raises the success rate from 33/51 (64.71%) to 37/51 (72.55%), a +7.84-point gain (+4 issues), while cutting total wall-clock time from 6h40m to 3h31m (1.9×) and average per-task time from 7.84 to 4.14 min. The speedup combines a shorter compressed context with a

Table 5: End-to-end SWE-bench agent evaluation with the OpenCode agent. Both configurations submit the same 51 issues and 20 concurrent requests; the average batch size reflects the concurrency that fits in memory.

Configuration	Success (↑)	Rate (%) (↑)	Total Time (↓)	Avg. Time/Task (↓)
OpenCode (no compression)	33/51	64.71	6h40m	7.84 min
LMI + OpenCode (3:1)	**37/51**	**72.55**	**3h31m**	**4.14 min**

higher achievable concurrency (average batch $3 \rightarrow 5$ under the same 20-task submission, a 66.7% throughput gain). Critically, the accuracy gain shows that compression also filters out the noise injected by BM25 over long contexts, letting the agent reason over cleaner evidence—improving success and efficiency simultaneously in a realistic agentic deployment.

4.5 Ablations

Our ablation studies isolate the contributions of semantic-aware structured compression, feature distillation, and latent memory interaction. For the latter, we train a 0.5B variant identical to LMI-Distill-0.5B except that the cross-window interaction module is removed; as reported in Appendix A.6, this causes a 26-point recall drop on *Needle5_in_Haystack_32K_atc* and pushes the 0.5B model below the 7B CPRS baseline, attributing the gains in Section 4.2.1 to latent interaction.

4.5.1 Ablation Study on Semantic-Aware Structured Compression

Removing the Semantic-Aware Structured (SAS) component while keeping all else unchanged consistently degrades performance across all seven QA benchmarks (Table 8, Appendix A.7): the average score drops from 29.95 to 28.70 for LMI-7B and from 29.36 to 27.63 for LMI-Distill-0.5B, confirming SAS's consistent benefit.

4.5.2 Ablation Study on Feature Distillation

To isolate feature distillation, we compare LMI-Distill-0.5B against a 0.5B model trained directly via supervised fine-tuning (LMI-SFT-0.5B). Distillation consistently outperforms direct SFT across both generators (Table 9, Appendix A.7), with a 2.44-point average gain under Qwen2.5-7B-Instruct (29.36 vs. 26.92) and a similar trend under Qwen2.5-14B-Instruct—confirming that distillation, rather than direct SFT, transfers the 7B model's compression capability to the 0.5B variant at a much smaller inference footprint.

5 Conclusion

We presented **LMI** (Latent Memory Interaction), a learned framework that introduces learnable latent memory tokens with cross-window interaction to establish global communication across context windows at near-linear complexity, combined with semantic-aware structured compression, perplexity-guided data generation, and feature distillation to a 0.5B model. Experiments show that LMI improves retrieval accuracy while retaining fewer sentences, maintains competitive QA performance under substantial compression, cuts serving latency by 8.7× on 128K workloads, and—in a real SWE-bench agent pipeline—raises task success by 7.84 points while cutting wall-clock time by 1.9×. LMI thus demonstrates that effective compression need not rely on local token importance alone: modeling information exchange across context regions via compact latent memories offers a promising direction for scalable, faithful context compression.

Reproducibility statement

All experiments use public benchmarks (NeedleBench, LongBench-v1, LoCoMo, and SWE-bench) and documented baselines; the data-construction procedure (Algorithm 1), model architectures, and hyperparameters are in the appendix. Code will be released upon acceptance.

AI USE STATEMENT

In this work, we used generative AI tools to assist with language polishing, translation, and revision during manuscript preparation. In addition, parts of the training, evaluation, and data-construction code were developed through interactive collaboration with AI coding agents. All core research contributions of this work—including the problem formulation, the LMI framework design, the algorithmic and architectural choices, and the experimental design—are entirely human-authored and did not involve generative AI tools. All AI-polished text was reviewed and revised by the authors, and all AI-assisted code was verified and tested for correctness by the authors before use. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Yushi Bai, Xin Lv, Jiajie Zhang, Yuze He, Ji Qi, Lei Hou, Jie Tang, Yuxiao Dong, and Juanzi Li. Longalign: A recipe for long context alignment of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 1376–1395, 2024a.

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

Nadezhda Chirkova, Thibault Formal, Vassilina Nikoulina, and Stéphane Clinchant. Provence: efficient and robust context pruning for retrieval-augmented generation. In *International Conference on Learning Representations*, volume 2025, pp. 38082–38106, 2025.

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

OpenCode Contributors. Opencode: An open-source ai coding agent. `https://github.com/sst/opencode`, 2024.

Ziqiang Cui, Yunpeng Weng, Xing Tang, Peiyang Liu, Shiwei Li, Bowei He, Jiamin Chen, Yansen Zhang, Xiuqiang He, and Chen Ma. Less is more: Elevating rag via performance-driven context compression. *arXiv preprint arXiv:2508.19282*, 2025.

Yuhong Dai, Jianxun Lian, Yitian Huang, Wei Zhang, Mingyang Zhou, Mingqi Wu, Xing Xie, and Hao Liao. Pretraining context compressor for large language models with embedding-based memory. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 28715–28732, 2025.

fpg2012. Cassum: A chinese long-text summarization dataset. `https://github.com/fpg2012/CASSum`, 2024.

Sensen Gao, Shanshan Zhao, Xu Jiang, Lunhao Duan, Yong Xien Chng, Qing-Guo Chen, Weihua Luo, Kaifu Zhang, Jia-Wang Bian, and Mingming Gong. Scaling beyond context: A survey of multimodal retrieval-augmented generation for document understanding. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4458–4489, 2026.

Suyu Ge, Xihui Lin, Yunan Zhang, Jiawei Han, and Hao Peng. A little goes a long way: Efficient long context training and inference with partial contexts. In *International Conference on Learning Representations*, volume 2025, pp. 6470–6483, 2025.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Curtis Hawthorne, Andrew Jaegle, Cătălina Cangea, Sebastian Borgeaud, Charlie Nash, Mateusz Malinowski, Sander Dieleman, Oriol Vinyals, Matthew Botvinick, Ian Simon, et al. General-purpose, long-context autoregressive modeling with perceiver ar. In *International Conference on Machine Learning*, pp. 8535–8558. PMLR, 2022.

Markus Hiller, Krista A Ehinger, and Tom Drummond. Perceiving longer sequences with bi-directional cross-attention transformers. *Advances in Neural Information Processing Systems*, 37:94097–94129, 2024.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. *arXiv preprint arXiv:2011.01060*, 2020.

Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International conference on machine learning*, pp. 4651–4664. PMLR, 2021.

Hanqi Jiang, Junhao Chen, Yi Pan, Ling Chen, Weihang You, Yifan Zhou, Ruidong Zhang, Yohannes Abate, and Tianming Liu. SYNAPSE: Empowering LLM agents with episodic-semantic memory via spreading activation. In *Findings of the Association for Computational Linguistics: ACL 2026*. Association for Computational Linguistics, 2026.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Longllmlingua: Accelerating and enhancing llms in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1658–1677, 2024.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. Swe-bench: Can language models resolve real-world github issues? In *International Conference on Learning Representations*, 2024.

Donghyeon Joo, Helya Hosseini, Ramyad Hadidi, and Bahar Asgari. Mustafar: Promoting unstructured sparsity for kv cache pruning in llm inference. *Advances in Neural Information Processing Systems*, 38:76342–76366, 2026.

Mo Li, Songyang Zhang, Taolin Zhang, Haodong Duan, Yunxin Liu, and Kai Chen. Needlebench: Evaluating llm retrieval and reasoning across varying information densities. *arXiv preprint arXiv:2407.11963*, 2024.

Yuankai Li, Jia-Chen Gu, Di Wu, Kai-Wei Chang, and Nanyun Peng. Brief: bridging retrieval and inference for multi-hop reasoning via compression. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 5464–5485, 2025a.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*, 2023.

Zongqian Li, Yixuan Su, and Nigel Collier. 500xcompressor: Generalized prompt compression for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 25081–25091, 2025b.

Dong Liu and Yanxuan Yu. Quickmerge++: Fast token merging with autoregressive prior. *arXiv preprint arXiv:2508.13204*, 2025.

Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. Generating wikipedia by summarizing long sequences. *arXiv preprint arXiv:1801.10198*, 2018.

Maxime Louis, Hervé Déjean, and Stéphane Clinchant. Pisco: Pretty simple compression for retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 15506–15521, 2025.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of llm agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13851–13870, 2024.

Tsendsuren Munkhdalai, Manaal Faruqui, and Siddharth Gopal. Leave no context behind: Efficient infinite context transformers with infini-attention. *arXiv preprint arXiv:2404.07143*, 2024.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, et al. Llmlingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 963–981, 2024.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009.

Weizhou Shen, Chenliang Li, Fanqi Wan, Shengyi Liao, Shaopeng Lai, Bo Zhang, Yingcheng Shi, Yuning Wu, Gang Fu, Zhansheng Li, et al. Qwenlong-cprs: Towards llms with dynamic context optimization. *arXiv preprint arXiv:2505.18092*, 2025a.

Weizhou Shen, Ziyi Yang, Chenliang Li, Zhiyuan Lu, Miao Peng, Huashan Sun, Yingcheng Shi, Shengyi Liao, Shaopeng Lai, Bo Zhang, et al. Qwenlong-l1. 5: Post-training recipe for long-context reasoning and memory management. *arXiv preprint arXiv:2512.12967*, 2025b.

GLM Team and Zhipu AI. Glm-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.

Fanqi Wan, Weizhou Shen, Shengyi Liao, Yingcheng Shi, Chenliang Li, Ziyi Yang, Ji Zhang, Fei Huang, Jingren Zhou, and Ming Yan. Qwenlong-l1: Towards long-context large reasoning models with reinforcement learning. *arXiv preprint arXiv:2505.17667*, 2025.

Yu Wang, Yifan Gao, Xiusi Chen, Haoming Jiang, Shiyang Li, Jingfeng Yang, Qingyu Yin, Zheng Li, Xian Li, Bing Yin, et al. Memoryllm: Towards self-updatable large language models. *arXiv preprint arXiv:2402.04624*, 2024.

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026a.

Peng Xu, Wei Ping, Xianchao Wu, Chejian Xu, Zihan Liu, Mohammad Shoeybi, and Bryan Catanzaro. Chatqa 2: Bridging the gap to proprietary llms in long context and rag capabilities. In *International Conference on Learning Representations*, volume 2025, pp. 67244–67258, 2025.

Zihao Xu, John Harvill, Ziwei Fan, Yizhou Sun, Hao Ding, and Hao Wang. Compressing sequences in the latent embedding space: k-token merging for large language models. *arXiv preprint arXiv:2604.15153*, 2026b.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*, 2018.

Song Yu, Xiaofei Xu, Ke Deng, Li Li, and Lin Tian. Tree of agents: Improving long-context capabilities of large language models through multi-perspective reasoning. *arXiv preprint arXiv:2509.06436*, 2025.

Shengbin Yue, Wei Chen, Siyuan Wang, Bingxuan Li, Chenchen Shen, Shujun Liu, Yuxuan Zhou, Yao Xiao, Song Yun, Xuanjing Huang, and Zhongyu Wei. Disc-lawllm: Fine-tuning large language models for intelligent legal services. *arXiv preprint arXiv:2309.11325*, 2023.

Peitian Zhang, Zheng Liu, Shitao Xiao, Ninglu Shao, Qiwei Ye, and Zhicheng Dou. Long context compression with activation beacon. In *International Conference on Learning Representations*, volume 2025, pp. 101932–101948, 2025.

Qingfei Zhao, Ruobing Wang, Yukuo Cen, Daren Zha, Shicheng Tan, Yuxiao Dong, and Jie Tang. Longrag: A dual-perspective retrieval-augmented generation paradigm for long-context question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 22600–22632, 2024.

A Appendix

A.1 Complexity and Inference

LMI is designed specifically for efficient long-context inference. Let N denote the original sequence length, K the chunk length, $M = N/K$ the number of chunks, and R the number of latent memory tokens per chunk. The context encoder processes each chunk independently for the majority of its computation. Therefore, the dominant self-attention cost is $O(NK)$, which becomes approximately linear in N when K is fixed. Cross-chunk communication only operates on $M * R$ latent representations and does not construct attention maps over the full N-token sequence. Moreover, our mean-based latent interaction introduces only $O(MR)$ additional aggregation cost per interaction layer.

The compression module is executed only during prefill. After compression, the target LLM receives the reduced sequence X_{comp} and performs standard autoregressive decoding. Consequently, the additional cost introduced by LMI is incurred once during context processing, while the reduced sequence length directly decreases the subsequent KV-cache footprint and decoding-time attention cost.

Overall, LMI provides a favorable trade-off between global context awareness and computational efficiency: latent memories enable information exchange across distant chunks, and semantic restructuring preserves coherent content. We complement this analysis with end-to-end wall-clock measurements (Table 4). In contrast to the quality evaluation in Section 4.2, which adopts Qwen2.5 generators to remain directly comparable with the evaluation configuration of the CPRS baseline, the efficiency study employs a larger mainstream open-source model (MiniMax-M2.7) to characterize the regime in which the cost of full-context serving is most pronounced. LMI operates on the input side and is agnostic to the downstream generator—the quality results already demonstrate that compressed contexts remain effective across different generators—and the attention and KV-cache overhead of full-context inference grows with generator scale, so the measured benefit reflects the deployment scenario where compression yields the greatest savings. The efficiency measurements above use the LMI-7B compressor; the distilled 0.5B variant incurs a substantially smaller per-token cost and would therefore only further reduce the compression overhead. This experiment focuses on a single representative serving configuration; a systematic study across context lengths, concurrency levels, and hardware platforms is left to future work.

A.2 Perplexity-Guided Key Sentence Mining

Training a context compression model requires fine-grained supervision indicating which portions of a long context are relevant to the downstream task. However, existing document–summary datasets

Algorithm 1 Perplexity-Guided Key Sentence Mining

Require: Source document $X = \{S_1, S_2, \ldots, S_N\}$
Ground-truth summary Y
Task query / instruction Q
Perplexity threshold τ
Ensure: Key sentence set E^*
$P_0 \leftarrow \text{PPL}(Y \mid Q)$
$G \leftarrow \emptyset$
for $i = 1, \ldots, N$ **do**
    $P_i \leftarrow \text{PPL}(Y \mid Q \oplus S_i)$
    $G_i \leftarrow P_0 - P_i$
    if $G_i \geq 0$ **then**
        $G \leftarrow G \cup \{(S_i, G_i)\}$
    end if
end for
Sort G in descending order of G_i
$E^* \leftarrow \emptyset$
for $(S_i, G_i) \in G$ **do**
    $P \leftarrow \text{PPL}(Y \mid Q \oplus E^* \oplus S_i)$
    $E^* \leftarrow E^* \cup \{S_i\}$
    if $P \leq \tau$ **then**
        break
    end if
end for
return E^*

typically provide only document-level supervision in the form of a source document paired with a ground-truth summary, without explicitly identifying the sentences that contribute to the target summary. Directly training a compressor from such weakly supervised data is therefore challenging, as the desired sentence-level relevance labels are latent.

We address this problem by introducing a Perplexity-Guided Key Sentence Mining procedure that automatically recovers sentence-level relevance from document–summary pairs. The algorithm workflow is illustrated in Algorithm 1. Our central observation is that a source sentence that contains information useful for generating the reference summary should improve a language model's ability to predict the summary. We therefore measure sentence relevance through the reduction in the perplexity (PPL) of the ground-truth summary after conditioning on the corresponding source sentence.

We vary α to control the perplexity threshold between the compressed text and the original full context, and employ adaptive compression to automatically determine the amount of text to retain. Across different α settings, the resulting compression ratios are consistently maintained at approximately 35%–40%. Despite this substantial reduction in context length, summaries generated from the compressed texts achieve ROUGE precision, recall, and F1 scores that are comparable to, and in some cases slightly higher than, those obtained from the full contexts. This observation demonstrates that the proposed data construction procedure effectively preserves the information relevant to summarization while substantially reducing the amount of input text, providing quantitative validation for the quality of the constructed dataset.

Through a series of comparative experiments, as is shown in Figure 4, we identify Qwen3-4B with $\alpha = 1.1$ as the final model and threshold configuration. This setting achieves a favorable balance between compression and summarization performance and is subsequently used for the construction of the training dataset.

A.3 Model Configurations and Training Hyperparameters

Table 6 summarizes the architectural and training configurations of the LMI compressor and its distilled variant.

2_0.5B	**P(↑)**	**R(↑)**	**F1(↑)**	**PPL(↓)**	**compress_Rate(↓)**
Full text	76.03	76.07	76.01	2.997	100%
α=1.05	78.43	77.19	77.71	3.037(+1.34%)	43.30%
α=1.10	78.84	78.16	78.43	3.165(+5.60%)	36.16%
2.5_7B	**P(↑)**	**R(↑)**	**F1(↑)**	**PPL(↓)**	**compress_Rate(↓)**
Full text	75.77	82.59	79.01	2.320	100%
α=1.05	77.64	82.65	80.03	2.373(+2.28%)	55.18%
α=1.10	77.70	82.66	80.07	2.470(+6.47%)	46.11%
2.5_14B	**P(↑)**	**R(↑)**	**F1(↑)**	**PPL(↓)**	**compress_Rate(↓)**
Full text	77.67	81.56	79.55	2.132	100%
α=1.05	78.62	81.36	79.94	2.181(+2.30%)	50.71%
α=1.10	78.68	81.10	79.84	2.269(+6.42%)	39.69%
3_4B	**P(↑)**	**R(↑)**	**F1(↑)**	**PPL(↓)**	**compress_Rate(↓)**
Full text	76.04	83.21	79.45	2.816	100%
α=1.05	77.86	83.88	80.73	2.861(+1.57%)	42.94%
α=1.10	78.53	83.79	81.04	2.966(+5.33%)	34.90%
3_8B	**P(↑)**	**R(↑)**	**F1(↑)**	**PPL(↓)**	**compress_Rate(↓)**
Full text	74.84	83.00	78.70	2.725	100%
α=1.05	77.47	83.60	80.38	2.769(+1.61%)	44.63%
α=1.10	78.12	83.36	80.61	2.877(+5.58%)	35.96%

Figure 4: Comparison of compression effects under multi-model and multi-parameter configuration conditions.

A.4 VISUALIZATION

As illustrated in Figure 5, we visualize the retrieval performance under the complex chained setting of *Needle5_in_Haystack_32K_atc*. LMI successfully retrieves all five target critical pieces of information, whereas CPRS only recovers three of them.

A.5 COMPARISON WITH THE CLOSELY RELATED COMPRESSOR

Since QwenLong-CPRS shares the window-parallel and hybrid causal/bidirectional design with LMI, we treat it as the closest system-level comparator in our study. Both compressors use 7B-parameter backbones initialized from Qwen2-7B-Base, ensuring a fair comparison at the same model scale and backbone. We note that the two systems also differ in training data and content-selection granularity, so this comparison characterizes the full LMI system against its closest neighbor rather than isolating the causal effect of any single component.

Table 6: Model configurations and training hyperparameters.

Configuration	Value
Backbone initialization	Qwen2-Base (7B / 0.5B)
Chunk length K	2048
Latent tokens per chunk R	48
Interaction layers	[3,7,11,15,19,21-27]
α_1 (mean weight, Eq. 5)	0.3
α_2 (max weight, Eq. 5)	0.7
Training epochs	3
Learning rate	1e-5
Batch size	128
Distillation teacher layers	-1

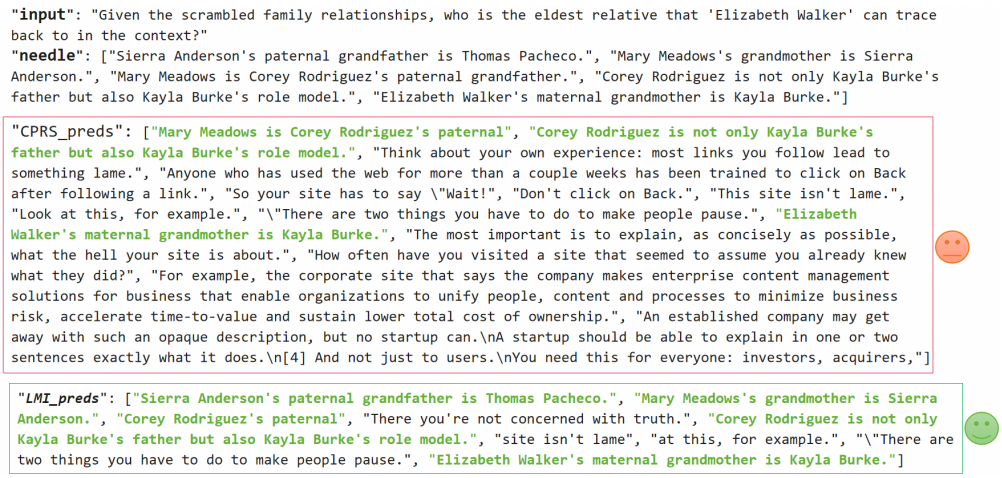

Figure 5: Performance Comparison between LMI-7B and CPRS on Needle5_in_Haystack_32K_atc.

At the evaluated operating points, LMI-7B improves recall over CPRS by 1.39%, 1.47%, and 34.40% on the *Needle5_in_Haystack_8K*, *32K*, and *32K_atc* settings, respectively, while retrieving fewer sentences (4.88/5.53/5.45 vs. 7.09/11.70/10.07 on average).

The contrast is most pronounced on *Needle5_in_Haystack_32K_atc*, where relevant evidence is distributed across a chain of relations and cannot be identified by window-local scoring alone: CPRS even falls below the RAG baseline (36.40 vs. 49.60), whereas LMI-7B reaches 70.80. This pattern is consistent with the intended benefit of cross-window communication—window-local importance estimation degrades precisely when relevance depends on evidence scattered across distant regions, which is the regime that latent interaction is designed to address—although it does not by itself attribute the improvement to that component.

A.6 Ablation on Latent Memory Interaction

To isolate the contribution of the latent memory interaction module, we train a dedicated 0.5B variant that is identical to LMI-Distill-0.5B in backbone (Qwen2-0.5B), training data, distillation recipe, and the semantic-aware structured compression (SAS) component, except that the cross-window latent interaction module is removed so that each window is compressed independently. We evaluate this variant on *Needle5_in_Haystack_32K_atc*, the most interaction-sensitive benchmark, where the relevant evidence is distributed across a chain of relations and cannot be recovered by window-local scoring alone.

Removing latent interaction causes a dramatic drop of 26.01 points in recall (60.60 $\rightarrow$ 34.59) and nearly doubles the number of retrieved sentences (7.68 $\rightarrow$ 14.21). Without cross-window communi-

Table 7: Ablation on latent memory interaction, evaluated on *Needle5_in_Haystack_32K_atc*. CPRS-7B and LMI-7B are included as reference points.

Model	Recall (%) ↑	Retrieved Sentence Num ↓
LMI-Distill-0.5B w/o interaction	34.59	14.21
CPRS-7B	36.40	10.07
LMI-Distill-0.5B	**60.60**	7.68
LMI-7B	**70.80**	**5.45**

cation, the model can no longer jointly integrate evidence scattered across distant chunks and instead resorts to retrieving more, less-relevant sentences. Notably, the interaction-free 0.5B model (34.59) even falls below the 7B-parameter CPRS baseline (36.40), whereas the interaction-equipped LMI-Distill-0.5B (60.60) surpasses CPRS-7B by 24.20 points. Flipping this single architectural switch thus moves a 0.5B-parameter compressor from below to above a 7B-parameter baseline, providing direct evidence that latent memory interaction is the decisive factor behind the strong retrieval gains reported in Section 4.2.1.

A.7 Ablation Tables

This section reports the full tables for the ablation studies in Section 4.4.

Table 8: Ablation study of the Semantic-Aware Structured (SAS) component in LMI on the QA task. All experiments use Qwen2.5-7B-Instruct as the generation model. "w/" and "w/o" denote using and removing SAS, respectively.

Model	SAS	NQ	Du	VC	Musi	QM	PC	TQ	AVG
LMI-7B	w/	**26.30**	**26.73**	**14.07**	**38.95**	**23.13**	**2.16**	**84.65**	**29.95**
	w/o	25.19	25.23	13.55	36.18	22.36	2.01	83.07	28.70
LMI-Distill-0.5B	w/	**25.22**	**27.90**	**14.23**	**31.26**	**23.49**	**4.86**	**86.86**	**29.36**
	w/o	23.23	26.52	12.18	28.92	22.13	4.21	84.25	27.63

Table 9: Ablation study of feature distillation for model compression. LMI-Distill-0.5B is trained via feature distillation, while LMI-SFT-0.5B is directly trained using supervised fine-tuning

Generation Model	Compression Model	NQ	DU	VC	Musi	QM	PC	TQ	AVG
Qwen2.5-7B-Instruct	LMI-Distill-0.5B	**25.22**	**27.90**	**14.23**	**31.26**	**23.49**	**4.86**	**86.86**	**29.36**
	Compression Ratio	14.41%	15.97%	15.32%	8.98%	27.85%	7.30%	10.75%	–
	LMI-SFT-0.5B	21.61	27.75	13.48	23.27	22.95	4.32	85.37	26.92
	Compression Ratio	14.37%	15.61%	15.27%	9.11%	28.29%	7.24%	10.75%	–
Qwen2.5-14B-Instruct	LMI-Distill-0.5B	**26.82**	**26.40**	**13.43**	**33.15**	**23.13**	**3.26**	**86.88**	**29.38**
	LMI-SFT-0.5B	23.95	25.29	12.63	31.55	22.45	1.67	85.40	27.84