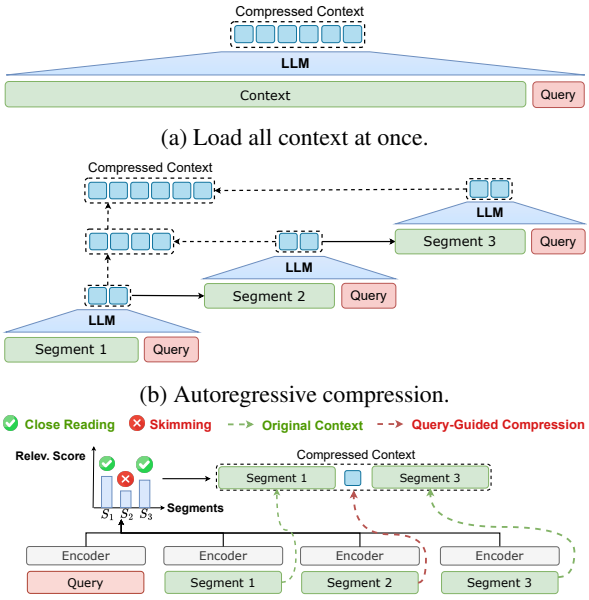
Read As Human: Compressing Context via Parallelizable Close Reading and Skimming

Anonymous ACL submission

Abstract

Large Language Models (LLMs) demonstrate exceptional capability across diverse tasks. However, their deployment in long-context scenarios is hindered by two challenges: computational inefficiency and redundant information. We propose **RAM** (**R**ead **A**s Hu**M**an), a context compression framework that adopts an adaptive hybrid reading strategy, to address these challenges. Inspired by human reading behavior (i.e., *close reading* important content while *skimming* less relevant content), RAM partitions the context into segments and encodes them with the input query *in parallel*. High-relevance segments are fully retained (*close reading*), while low-relevance ones are query-guided compressed into compact summary vectors (*skimming*). Both explicit textual segments and implicit summary vectors are concatenated and fed into decoder to achieve both superior performance and natural language format interpretability. To refine the decision boundary between close reading and skimming, we further introduce a contrastive learning objective based on positive and negative query–segment pairs. Experiments demonstrate that RAM outperforms existing baselines on multiple question answering and summarization benchmarks across two backbones, while delivering up to a 12× end-to-end speedup on long inputs (average length 16K; maximum length 32K).



(a) Load all context at once.

(b) Autoregressive compression.

(c) RAM.

Figure 1: Comparison of RAM with other task-aware context compression methods. Existing task-aware methods either require loading the entire input sequence at once for compression (Figure 1a) or rely on autoregressive compression (Figure 1b), both of which suffer from computational inefficiency. In contrast, RAM processes all segments and the query *in parallel* and adaptively decides (based on relevance) which segments to *close reading* and which to *skim* (Figure 1c).

1 Introduction

The rise of Large Language Models (LLMs) has profoundly reshaped the paradigm of Natural Language Processing (NLP), demonstrating remarkable generalization across a wide range of tasks (Yang et al., 2024; DeepSeek-AI et al., 2025; Team et al., 2025). However, with the widespread adoption of prompt engineering techniques such as Retrieval-Augmented Generation (RAG) (Lewis et al., 2020), input prompts are increasingly long, even exceeding tens of thousands of tokens. Deploying LLMs in such long-context scenarios faces two challenges: (1) Computational inefficiency. Mainstream models (e.g., Qwen, DeepSeek) rely on the Transformer architecture (Vaswani et al., 2017), whose self-attention mechanism incurs quadratic time complexity with respect to input sequence length. (2) Information redundancy. Natural language is inherently redundant (Shannon, 1948), and this redundancy is exacerbated in long context, thereby degrading performance (Jiang et al., 2024; Ge et al., 2024; Tang et al., 2025a,b).

Context compression has emerged as a promis-

ing direction to address these challenges by substantially reducing sequence length while filtering out irrelevant content. Existing methods can fall into two categories: task-agnostic context compression methods (Liao et al., 2025; Li et al., 2025; Dai et al., 2025; Choi et al., 2025; Zhang et al., 2025; Tang et al., 2025b) and task-aware context compression methods. Task-agnostic methods lack query guidance and are prone to losing key information (relevant to query). In contrast, task-aware methods leverage the query to guide compression and better preserve key information. However, existing task-aware methods face two key challenges: (1) Low computational efficiency. As shown in Figure 1, existing methods either process the full long context in at once (Cao et al., 2024; Hwang et al., 2025a; Zhao et al., 2025; Liskavets et al., 2025; Fang et al., 2025), incurring highly inefficient computation on long sequences, or rely on autoregressive compression (Yoon et al., 2024; Jiang et al., 2024; Tang et al., 2025a), which also limits computational efficiency. (2) A trade-off between key information retention and interpretability. Some methods *directly prune* low relevance segments or tokens (Yoon et al., 2024; Jiang et al., 2024; Hwang et al., 2025a; Tang et al., 2025a; Zhao et al., 2025; Liskavets et al., 2025; Fang et al., 2025), risking loss of key information, while others compress the context into implicit semantic vectors (Cao et al., 2024), sacrificing the interpretability of the natural language format. This motivates the following research question: *How can we achieve efficient context compression that preserves as much key information as possible while remaining natural language format interpretability?*

To address these limitations, we draw inspiration from cognitive science and model human reading behavior: when reading, humans typically perform *close reading* on content highly relevant to their current goal, while adopting a *skimming* strategy for background information (Duggan and Payne, 2011; Wolf, 2018). Close reading focuses on the full structural and semantic details of the original context, thereby supporting deep comprehension. In contrast, skimming extracts key information and aggressively discards less relevant content, significantly reducing cognitive load while still capturing the query-relevant semantic gist. Motivated by this, we propose **RAM** (**R**ead **A**s Hu**M**an), which formalizes context compression as an efficient and adaptive hybrid reading strategy. Specifically, RAM first partitions a long context into multiple segments and processes all segments together with the input query *in parallel*, *avoiding the efficiency bottlenecks of existing methods that either load the entire context at once or rely on autoregressive compression*. Subsequently, RAM adaptively compresses the context based on query-segment relevance: highly relevant segments are fully retained (i.e., *close reading*) to ensure key information is preserved in natural language format, while less relevant segments are compressed via a query-guided mechanism into compact, implicit summary vectors (i.e., *skimming*), capturing only the query-relevant semantic gist while drastically reducing less relevant content. Then, RAM concatenates the explicit *close reading* segments with the implicit *skimming* summary vectors to form a hybrid contextual representation, which is fed into the decoder. *This design not only preserve as much key information as possible but also maintains natural language format interpretability.* We further introduce a contrastive learning objective that leverages annotated positive and negative query-segments pairs, optimizing the RAM's ability to distinguish between *close reading* and *skimming*, which enables RAM to more faithfully emulate human-like adaptive reading.

Our main contributions are threefold: (1) We propose RAM, an efficient and interpretable compression framework that combines explicit and implicit representations. By parallelizing segment encoding with the query and applying differentiated treatment based on relevance, RAM avoids the inefficiencies of full-sequence loading or autoregressive compression while maintaining interpretability. (2) We introduce a contrastive learning object for more adaptive compression. By modeling query-segment relevance as a contrastive task, the model learns a more robust decision boundary between *close reading* and *skimming*. (3) RAM achieves superior performance across multiple QA and summarization benchmarks under various compression budgets across two backbones, offers approximately 12$\times$ end-to-end speedup on long inputs (average length 16K; maximum length 32K).

2 Related Work

Task-Agnostic Context Compression Methods. Task-agnostic methods compress input context without relying on specific queries, preserving global semantics to support diverse downstream tasks. This category of work can further fall into

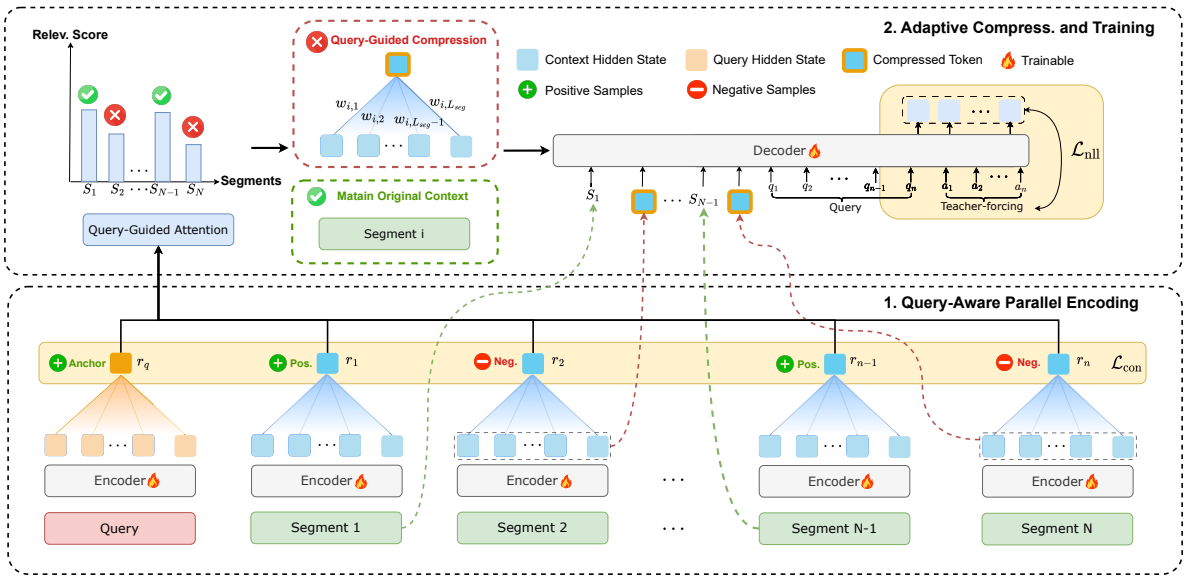


Figure 2: Overview of the RAM framework. The framework consists of two main stages: (1) **Query-Aware Parallel Encoding**: The query and segmented context are encoded *in parallel*. A query-guided attention mechanism computes a relevance score for each segment, determining whether to retain it as original text (*close reading*) or compress it into a compact vector (*skimming*), with the number of segments to retain derived from the compression ratio via Eq. (3). (2) **Adaptive Compression and Training**: The retained text and compressed vectors are fed into a decoder to produce the final output. The model is trained end-to-end using a language modeling objective $\mathcal{L}_{\text{nll}}$ and a contrastive learning objective $\mathcal{L}_{\text{con}}$ to jointly optimize overall performance.

two categories: (1) Hard prompt compression, which retains explicit textual tokens. Representative methods include metric-based pruning using information entropy (Li et al., 2023; Jiang et al., 2023) or bidirectional semantics (Pan et al., 2024), and summarization-based approaches (Xu et al., 2024) that train a query-agnostic summarizer. (2) Soft prompt compression, which maps context into non-lexical soft embeddings. Representatives include encoder-decoder frameworks (Ge et al., 2024; Cheng et al., 2024; Rau et al., 2024; Tan et al., 2024; Liao et al., 2025; Li et al., 2025; Dai et al., 2025; Choi et al., 2025; Tang et al., 2025b), attention mask modification (Mu et al., 2023; Petrov et al., 2025; Ye et al., 2025), and autoregressive modeling (Chevalier et al., 2023; Zhang et al., 2025; Deng et al., 2025) that treats compression as sequence generation conditioned on previously compressed tokens. *Despite preserving general semantics, these methods prone to discard task-relevant information, degrading performance due to the lack of query awareness.*

Task-Aware Context Compression Methods. Task-aware methods incorporate the query during compression to retain task-relevant content. Representative methods include generating query-guided summaries (Yoon et al., 2024; Hwang et al., 2025a), merging query-weighted tokens (Cao et al., 2024), or pruning low query relevance tokens (Jiang et al., 2024; Tang et al., 2025a; Liskavets et al., 2025; Fang et al., 2025; Zhao et al., 2025). *While preserving task-specific content, these methods either require loading the full sequence at once or depend on iterative autoregressive compression (both significantly hindering efficiency) and face an inherent trade-off between preserving key information and maintaining natural language format interpretability.*

3 Method

We propose RAM, a novel context compression framework that mimics human reading behavior: close reading highly relevant segments while skimming over less relevant ones via query-guided compression. The method operates within an encoder–decoder architecture and consists of two core stages: (1) query-aware parallel encoding of context segments, and (2) adaptive compression and training. Below, we detail each component.

3.1 Query-Aware Parallel Encoding

Given a query Q and a long context $\mathbf{C}$ segmented into N *equal-length* chunks $\{S_1, S_2, \ldots, S_N\}$., RAM processes all segments *in parallel* with the query using a shared encoder, avoiding the

quadratic cost of full-sequence attention and iterative compression.

Parallel Encoding. The original input sequence $\{S_1, S_2, \ldots, S_N, Q\}$ is passed through a trainable LLM-based encoder (e.g., LLaMA or Qwen) *in parallel* to obtain contextualized hidden states. Let $\mathbf{H}$ denote the last hidden states.

Representative Tokens for the Query and Segments. We construct a compact representation for each text sequence by mean-pooling its last hidden states. For segment S_i with length L_{seg}, its representation is computed as

$$\mathbf{r}_i = \frac{1}{L_{\text{seg}}} \sum_{j=1}^{L_{\text{seg}}} \mathbf{h}_{i,j}, \tag{1}$$

where L_{seg} refers to the token length of each segment; $\mathbf{h}_{i,j}$ denotes the last hidden state of the j-th token in S_i. The query representation $\mathbf{r}_q$ is obtained in the same way by mean-pooling the last-layer hidden states of the query tokens.

3.2 Adaptive Compression and Training

Based on relevance to the query, RAM dynamically decides whether to preserve each segment verbatim ("close reading") or compress it into a single vector ("skimming"), guided by a learnable selection mechanism.

Query-Guided Attention. We compute cosine similarity between $\mathbf{r}_q$ and each $\mathbf{r}_i$, followed by a softmax to obtain segment probabilities p_i:

$$p_i = \frac{\exp\left(\mathbf{r}_q^\top \mathbf{r}_i / \tau\right)}{\sum_j \exp\left(\mathbf{r}_q^\top \mathbf{r}_j / \tau\right)}. \tag{2}$$

Given a sampled compression rate α (i.e., sampled from $\{2, 4, 8, 16, 32\}$), we determine the number of segments k to fully preserve. k satisfies:

$$k = \lfloor \frac{L_{\text{org}}}{\alpha L_{\text{seg}}} \rfloor, \tag{3}$$

where L_{org} is the token length of original input. We select the top-k segments with highest p_i for retention; others are compressed.

Query-Guided Compression (Skimming). For segments marked for skimming, we compute a query-aware weighted average of token hidden states:

$$\begin{aligned} w_{i,t} &= \text{softmax}_t\big(\cos(\mathbf{h}_{i,t}, \mathbf{r}_q)\big), \\ \mathbf{c}_i &= \sum_{t=1}^{L_{\text{seg}}} w_{i,t}\, \mathbf{h}_{i,t}. \end{aligned} \tag{4}$$

where $\cos(\cdot,\cdot)$ denotes cosine similarity. This yields a single embedding $\mathbf{c}_i \in \mathbf{R}^d$ per skimmed segment, where d is the hidden dimension.

Hybrid Compressed Representation Construction. Let $\mathcal{K} \subseteq \{1, \ldots, N\}$ denote the set of segment indices selected for close reading, and let $\bar{\mathcal{K}}$ be its complement. For each segment i, we construct its contribution to the compressed context representation as

$$\tilde{\mathbf{M}}_i = \mathbf{I}_{\{i \in \mathcal{K}\}} \cdot \mathbf{e}(S_i) + \mathbf{I}_{\{i \in \bar{\mathcal{K}}\}} \cdot \mathbf{W}_{\text{align}} \mathbf{c}_i, \tag{5}$$

where $\mathbf{e}(\cdot)$ denotes the word embedding lookup in the Decoder LLM; $\mathbf{c}_i \in \mathbf{R}^d$ is the query-guided compressed vector obtained via query-guided compression (skimming), and $\mathbf{W}_{\text{align}} \in \mathbf{R}^{d \times d}$ is a trainable semantic alignment matrix applied only to skimmed segments to bridge the representation gap between explicit and implicit forms. The indicator $\mathbf{I}_{\{\cdot\}}$ selects the appropriate representation path per segment. The final compressed context is then formed by concatenating all segment representations in their original order:

$$\mathbf{M} = \left[\tilde{\mathbf{M}}_1; \tilde{\mathbf{M}}_2; \ldots; \tilde{\mathbf{M}}_N\right] \in \mathbf{R}^{L_{\text{c}} \times d}, \tag{6}$$

where $L_{\text{c}} = \sum_{i \in \mathcal{K}} L_{\text{seg}} + |\bar{\mathcal{K}}|$ is the total length after compression. This hybrid representation preserves verbatim tokens for high-relevance segments while replacing low-relevance ones with compact, query-guided skimming, thereby achieving both fidelity and efficiency.

Training Objective. The compressed memory $\mathbf{M}$ is concatenated with the query embeddings and answer tokens, and fed into the decoder for standard teacher-forced language modeling. Let $\mathbf{Y} = [y_1, \ldots, y_{N_a}]$ denotes the answer token sequence. The language modeling loss is

$$\mathcal{L}_{\text{nll}} = -\frac{1}{N_a} \sum_{t=1}^{N_a} \log P(y_t \mid y_{<t}, \mathbf{M}, Q), \tag{7}$$

where $P(y_t \mid \cdot)$ is derived from the softmax-normalized logits, and loss is computed only over answer positions.

To improve relevance discrimination, we introduce a contrastive loss during training. Given ground-truth positive/negative segment labels (e.g., from answer span annotations), we encourage higher similarity between the query and positive

segments, and lower similarity with negative ones:

$$\mathcal{L}_{\text{con}} = -\frac{1}{|\mathcal{P}|} \sum_{i \in \mathcal{P}} \log \frac{\exp(\cos(\mathbf{r}_q, \mathbf{r}_i)/\tau)}{\sum_{j=1}^{N} \exp(\cos(\mathbf{r}_q, \mathbf{r}_j)/\tau)}, \tag{8}$$

where $\mathcal{P}$ is the set of positive segment indices and τ is a temperature parameter.

The total training objective combines both terms:

$$\mathcal{L}_{\text{RAM}} = \mathcal{L}_{\text{nll}} + \mathcal{L}_{\text{con}}. \tag{9}$$

This design enables RAM to achieve high compression efficiency, strong task performance, and natural language format interpretability.

4 Experiment

In this section, we aim to address the following four research questions (RQs): (1) How does RAM perform compared to baseline methods (RQ1)? (2) How efficient is RAM (RQ2)? (3) How efficient and robust is RAM under various compression ratios in long-context scenarios (RQ3)? (4) How effective is each component within RAM (RQ4)?

4.1 Settings

Training. RAM requires *only a single training* to support diverse downstream tasks under various compression ratios, including question answering and summarization. We randomly sample 20,000 examples from each of HotpotQA (Yang et al., 2018), 2WikiMQA (Ho et al., 2020), NaturalQuestions (Liu et al., 2024), NarrativeQA (Kociský et al., 2018), and MultiNews (Fabbri et al., 2019) to construct the final training set (more details about datasets in Appendix C). All training samples are truncated to a maximum token length of 20K. During training, we use a batch size of 32 and a learning rate of 1×10^{-5} with a linear learning rate decay schedule. We set the segment size to 50 to enable finer-grained control over compression; in practice, the segment size has minimal impact on performance (see Appendix A). Following Chen et al. (2020), τ is set to 0.1. The training paradigm is illustrated in Figure 2. For each training sample, we randomly sample a compression rate from the $\{2, 4, 8, 16, 32\}$ to cover various compression situations. We set the maximum test length for NarrativeQA to 32K, resulting in a test set with an average length of 16K.

Contrastive Learning Data Processing. We convert ground truth positions from NaturalQuestions, HotpotQA, and 2WikiMQA into segments. MultiNews is excluded as it is a summarization dataset. For NarrativeQA (which lacks position annotations), we segment documents and use Qwen3-235B-A22B-Instruct to label segments as answer-containing or not (see Appendix B).

Implementation Details. Our implementation is based on LLaMA-3.1-8B (Instruct) and Qwen3-4B (Instruct). To ensure a fair comparison, all baseline results are reproduced using the officially released code. All experiments are conducted on 8 GPUs with compute performance comparable to NVIDIA H800, using the Hugging Face framework.

Evaluation Metrics. Following Hwang et al. (2025b), for question answering tasks, we report Exact Match (EM) and F1 score (Yang et al., 2018). For summarization on MultiNews, performance is measured using the F1 score.

Baselines. We conduct comprehensive comparisons against both task-specific and task-agnostic text compression methods. Specifically, we include task-specific approaches (e.g., LongLLMLingua (Jiang et al., 2024), Provence (Chirkova et al., 2025), EXIT (Hwang et al., 2025b)) and task-agnostic approaches (e.g., LLMLingua-2 (Pan et al., 2024), Activation Beacon (Zhang et al., 2025)). We compare with Activation Beacon under the setting using LLaMA-3.1-8B-Instruct as the backbone, as its open-source code only supports LLaMA-3.1-8B-Instruct. Additionally, we also report model performance under the settings of the original (uncompressed) prompt and zero-shot prompting.

4.2 Main Results (RQ1)

As shown in Table 1, RAM demonstrates significant advantages across multiple benchmarks. We summarize our findings as follows: (1) **Performance Superiority.** Under both 4× and 8× compression constraints, RAM consistently outperforms existing baselines on all benchmarks, achieving state-of-the-art results (bolded) on EM and F1 metrics most of the time. This indicates its superior capability in preserving semantics and supporting question answering under long-context compression scenarios. (2) **Robustness.** RAM maintains stable and strong performance across different backbone models (*i.e.*, LLaMA-3.1-8B-Instruct and Qwen3-4B-Instruct) and compression constraint (*i.e.*, 4× and 8×). This confirms the adaptability of RAM to diverse model architec-

Table 1: Experimental results on four QA benchmarks and the MultiNews summarization benchmark. Closed-book indicates using only the input question as the input, while Original Prompt indicates using original context as the input. We **bold** the optimal results.

Methods	NaturalQA		2WikiMQA		HotpotQA		NarrativeQA		MultiNews	AVG	
	EM	F1	EM	F1	EM	F1	EM	F1	F1	EM	F1
LLaMA-3.1-8B-Instruct											
Closed-book	14.05	21.98	17.14	23.42	10.77	19.21	1.03	9.00	-	10.75	18.40
Original Prompt	37.63	48.25	26.07	39.60	26.81	45.86	6.39	19.26	34.79	24.22	37.55
4x Compression Constraint											
ICAE	35.82	37.57	32.66	37.78	26.18	36.17	4.70	12.46	28.16	24.84	30.43
LLMLingua-2-large	29.98	43.92	18.16	32.76	29.83	45.25	12.59	24.93	**30.08**	22.64	35.39
Activation Beacon	47.04	58.97	36.69	44.20	43.78	57.44	18.61	28.29	28.52	36.53	43.48
Provence	37.25	49.13	30.25	45.87	40.57	57.51	15.13	30.11	30.14	30.80	42.55
EXIT	43.31	57.70	28.43	42.25	45.55	58.81	11.00	24.17	29.71	32.07	42.53
LongLLMLingua	45.50	58.34	24.58	35.30	34.88	51.40	7.15	19.17	26.32	28.03	38.11
RAM	**65.35**	**59.22**	**48.66**	**54.89**	**46.24**	**59.13**	**19.64**	**32.47**	29.21	**44.97**	**46.98**
8x Compression Constraint											
ICAE	36.13	38.31	33.42	37.78	26.92	35.77	4.41	12.41	**27.92**	25.22	30.44
LLMLingua-2-large	20.11	33.58	14.05	27.62	21.29	34.55	11.28	22.46	27.55	16.68	29.15
Activation Beacon	41.21	55.09	35.53	42.98	36.29	48.57	17.95	26.80	25.19	32.74	39.73
Provence	34.16	47.19	28.81	43.12	37.22	49.91	14.10	29.15	26.19	28.57	39.11
EXIT	43.88	55.42	22.94	33.33	32.16	50.19	6.86	18.62	27.15	26.46	36.94
LongLLMLingua	39.50	54.30	20.33	30.13	28.63	44.10	4.52	15.84	21.56	23.24	33.19
RAM	**62.41**	**57.14**	**39.63**	**45.24**	**38.82**	**50.80**	**18.80**	**31.57**	26.11	**39.92**	**42.17**
Qwen3-4B-Instruct											
Closed-book	10.17	17.75	13.67	23.92	12.16	20.45	0.47	9.65	-	9.12	17.94
Original Prompt	32.77	44.44	32.94	43.60	44.33	60.47	8.55	20.06	32.09	29.65	40.13
4x Compression Constraint											
ICAE	18.97	20.05	25.91	28.52	18.41	25.34	2.53	11.72	22.78	16.45	21.68
LLMLingua-2-large	23.09	35.72	25.17	31.58	28.20	41.02	7.89	19.03	29.56	21.09	31.38
Provence	31.11	43.39	39.52	48.32	42.38	56.11	11.00	24.46	28.30	31.00	40.12
EXIT	40.00	52.86	37.06	45.04	45.24	58.13	5.64	15.59	31.86	31.99	40.70
LongLLMLingua	40.23	53.01	26.81	32.68	30.13	42.86	2.82	12.20	24.15	25.00	32.98
RAM	**66.59**	**59.97**	**50.39**	**56.47**	**46.37**	**59.28**	**19.45**	**31.92**	**32.07**	**45.70**	**47.94**
8x Compression Constraint											
ICAE	19.49	19.71	26.07	28.55	18.08	25.84	2.45	11.75	23.42	16.52	21.85
LLMLingua-2-large	15.37	26.67	21.37	25.57	18.60	28.23	6.20	15.99	26.41	15.39	24.57
Provence	31.30	43.01	37.25	44.72	36.74	48.71	10.71	24.34	24.47	29.00	37.05
EXIT	41.54	53.90	29.80	35.59	35.75	48.40	2.82	11.53	28.07	27.48	35.50
LongLLMLingua	31.41	45.27	23.65	27.91	24.45	35.51	1.88	9.76	20.48	20.35	27.79
RAM	**61.09**	**56.15**	**40.24**	**45.66**	**37.90**	**49.15**	**19.92**	**31.22**	**28.50**	**39.79**	**42.14**

tures and compression strategies. (3) **Length Extrapolation Capability**: Although trained with a maximum input length of 20K tokens, RAM outperforms both the Original Prompt and other baselines on NarrativeQA (with the max length up to 32K tokens), demonstrating not only effective context compression but also the ability to maintain semantic coherence and inference accuracy on substantially longer inputs, highlighting its strong generalization and extrapolation capacity. This extrapolation behavior is particularly promising for real-world applications where input lengths often exceed training regimes, suggesting that RAM learns compositional representations rather than memorizing fixed-length patterns.

4.3 Efficiency Analysis (RQ2)

We analyze the computational efficiency of the RAM framework. By segmenting the long context and encoding each segment with the query *in parallel* (followed by "skimming" low-relevance segments via lightweight compression). RAM *fundamentally avoids* the bottlenecks of conventional approaches that either process the full input in one shot or rely on autoregressive iteration. This design drastically reduces inference cost. The entire pipeline consists of two stages: (1) compression, which includes query-aware parallel encoding and

adaptive compression, and (2) decoding, whose floating-point operations (FLOPs) can be modeled separately.

Compression Stage. This stage involves two core components. First, the original context $\mathbf{C}$ of length L_{org} is split into N segments, each encoded *in parallel* alongside the query Q (length L_q). Second, a query-guided attention mechanism computes a relevance score for each segment to decide whether to retain it as-is ("close reading") or compress it into a compact vector ("skimming"). Compressed segments are aggregated via a lightweight operation (e.g., weighted average) into a single summary vector. Let L_c denote the total effective length fed to the decoder (i.e., sum of retained token lengths and number of skimmed vectors), and $\alpha = L_c/L_{\text{org}}$ the compression ratio. The FLOPs of this stage are:

Table 2: Latency evaluation on NarrativeQA (average length 16K; maximum length 32K) using Qwen3-4B as backbone. Each compression method's total latency can be divided into compression latency and inference latency.

Methods	Compression Constraint				
	2x	4x	8x	16x	32x
Compression Latency (s)					
EXIT	303.61	300.86	301.30	302.45	302.06
Provence	2.12	2.12	2.12	2.13	2.11
LongLLMLingua	24.30	10.69	6.96	5.25	4.66
RAM	**0.10**	**0.09**	**0.09**	**0.09**	**0.08**
Inference Latency (s)					
EXIT	0.96	0.81	0.75	0.68	0.56
Provence	0.92	0.83	0.77	0.71	0.64
LongLLMLingua	0.87	0.68	0.65	0.59	0.48
RAM	**0.33**	**0.20**	**0.15**	**0.13**	**0.12**
End-to-End Latency (s)					
EXIT	304.57	301.67	302.05	303.13	302.62
Provence	3.04	2.95	2.89	2.84	2.75
LongLLMLingua	25.17	11.37	7.61	5.84	5.14
RAM	**0.43**	**0.29**	**0.24**	**0.22**	**0.20**
Original Prompt			1.23		

$$\text{FLOPs}_{\text{comp}} = F_{\text{ParaEnc}}(Q, \mathbf{C}) + F_{\text{QueryAttn}}(Q, \mathbf{R}), \tag{10}$$

where $F_{\text{ParaEnc}}(Q, \mathbf{C})$ denotes the cost of parallel encoding. Since each segment has length L_{seg} and is processed independently, its complexity is $O(N \cdot L_{\text{seg}}^2) = O(L_{\text{seg}} \cdot L_{\text{org}})$, which is substantially lower than the $O(L_{\text{org}}^2)$ cost of full-sequence encoding (note $N \cdot L_{\text{seg}} = L_{\text{org}}$ and $L_{\text{seg}} \ll L_{\text{org}}$). Meanwhile, $F_{\text{QueryAttn}}(Q, \mathbf{R})$ (i.e., the cost of query-guided attention over N relevance scores) incurs only $O(N)$ computational overhead, which is linear in the number of segments.

Decoding Stage. Assuming the generated answer has length L_a, decoding requires L_a forward passes. The FLOPs of the i-th pass depend on the compressed context length L_c and query length L_q:

$$\text{FLOPs}_i^{\text{forward}} = F_{\text{Decoder}}(\mathbf{M}, Q, i), \tag{11}$$

where $\mathbf{M}$ denotes the mixed input (retained tokens and skimmed vectors). The total FLOPs are therefore:

$$\text{FLOPs} = \sum_{i=1}^{L_a} \text{FLOPs}_i^{\text{forward}} + \text{FLOPs}_{\text{comp}}. \tag{12}$$

Empirical results show that under the same 32× compression constraint, RAM achieves significantly lower end-to-end latency compared to task-relevant baselines (see Table 2).

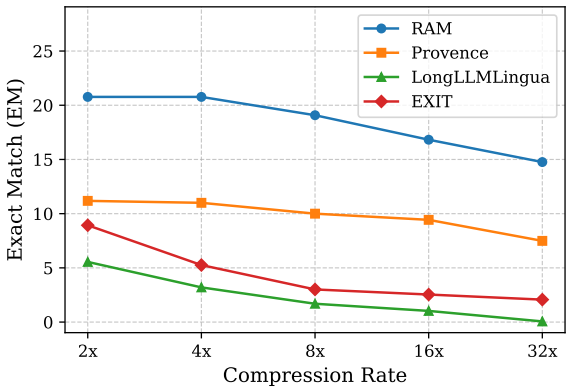


Figure 3: Performance under different compression rates on NarrativeQA. All methods use Qwen3-4B as the backbone.

4.4 Robustness Across Compression Rates (RQ3)

As shown in Figure 3, RAM consistently maintains a clear advantage across different compression rates (from 2x to 32x), indicating that RAM can effectively adapt to varying compression levels via a single run and exhibits strong robustness. In contrast, baseline methods such as Provence, LongLLMLingua, and EXIT exhibit a steady decline in Exact Match (EM) scores as compression becomes more aggressive, suggesting their sensitivity to high compression ratios. Notably, while all methods use Qwen3-4B as the backbone, RAM's performance remains relatively stable, demonstrating its resilience and practical utility for real-world applications requiring dynamic compression.

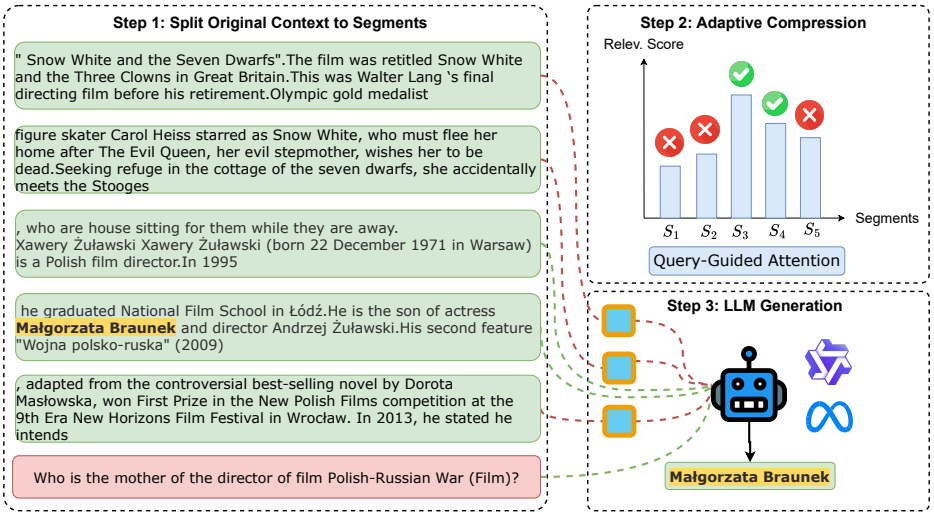


Figure 4: A case study from 2WikiMQA dataset.

Table 3: Ablation study on NaturalQuestions, 2WikiMQA under 8x compression constraint using Qwen3-4B as backbone.

Methods	NaturalQA		2WikiMQA	
	EM	F1	EM	F1
Default	**62.41**	**57.14**	**39.63**	**45.24**
w/o Skimming	58.38	53.95	37.19	42.10
w/o Close Reading	46.40	45.97	34.28	39.31
w/ AP Skimming	59.54	54.10	37.21	42.30
w/o Contrast Learning	49.27	46.03	33.29	37.42

4.5 Ablation Study (RQ4)

To address RQ4, we conduct three ablation studies using Qwen3-4B as the backbone model to examine the contribution of each component in RAM to its overall performance: (1) The *w/o* Skimming variant discards segments marked for compression after parallel encoding and adaptive selection, without generating any compressed tokens. (2) The *w/o* Close Reading variant uniformly applies skimming to all segments. (3) The *w/* AP Skimming variant replaces the query-guided skimming compression for each segment with simple Average Pooling (AP). (4) The *w/o* Contrastive Learning variant removes the contrastive learning term from the total training objective. Removing any component leads to a clear drop of the metric, demonstrating the necessity and effectiveness of each component. Discarding compressed tokens inevitably results in greater loss of key information, thereby degrading performance. Replacing query-aware compression with average pooling diminishes RAM's sensitivity to query-relevant content within compressed segments, diluting key information in the compressed representation. Removing the contrastive loss weakens the model's ability to distinguish between segments that require close reading or skimming, potentially leading to over-compression of important content and consequent performance degradation.

4.6 Case Study

RAM computes the relevance of each segment to the query using query-guided attention and applies close reading to highly relevant segments and skimming to less relevant ones. As shown in Figure 4, for the query "Who is the mother of the director of film *Polish-Russian War*?" (the answer is *Małgorzata Braunek*), segments S_3 and S_4 are processed with *close reading* due to their high relevance, while the remaining segments undergo *skimming*.

5 Conclusion

Inspired by human reading strategies, we propose RAM, a task-aware parallel context compression framework. RAM adaptively compresses query-irrelevant context segments (skimming) while preserving important parts (close reading). We further introduce a contrastive objective over the query and multiple context segments to better distinguish regions needing close reading from those suitable for skimming, improving compression quality. Extensive experiments demonstrate that RAM consistently outperforms strong baselines, delivering better effectiveness with substantial efficiency gains.

Limitations

Although RAM demonstrates strong performance across various long-context benchmarks and compression rates, along with high efficiency and good interpretability, it has certain limitations. Specifically, the data labels used for contrastive learning on NarrativeQA are generated by the Qwen3-235B-A22B-Instruct. While this model provides high-precision labels, they are not guaranteed to be fully accurate. Nevertheless, as shown in Table 3, contrastive learning based on these labels remains effective.

References

Zhiwei Cao, Qian Cao, Yu Lu, Ningxin Peng, Luyang Huang, Shanbo Cheng, and Jinsong Su. 2024. Retaining key information under high compression ratios: Query-guided compressor for llms. In *ACL (1)*, pages 12685–12695. Association for Computational Linguistics.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A simple framework for contrastive learning of visual representations. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

Xin Cheng, Xun Wang, Xingxing Zhang, Tao Ge, Si-Qing Chen, Furu Wei, Huishuai Zhang, and Dongyan Zhao. 2024. xrag: Extreme context compression for retrieval-augmented generation with one token. In *NeurIPS*.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. 2023. Adapting language models to compress contexts. In *EMNLP*, pages 3829–3846. Association for Computational Linguistics.

Nadezhda Chirkova, Thibault Formal, Vassilina Nikoulina, and Stéphane Clinchant. 2025. Provence: efficient and robust context pruning for retrieval-augmented generation. In *ICLR*. OpenReview.net.

Eunseong Choi, June Park, Hyeri Lee, and Jongwuk Lee. 2025. Conflict-aware soft prompting for retrieval-augmented generation. *CoRR*, abs/2508.15253.

Yuhong Dai, Jianxun Lian, Yitian Huang, Wei Zhang, Mingyang Zhou, Mingqi Wu, Xing Xie, and Hao Liao. 2025. Pretraining context compressor for large language models with embedding-based memory. In *ACL (1)*, pages 28715–28732. Association for Computational Linguistics.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948.

Chenlong Deng, Zhisong Zhang, Kelong Mao, Shuaiyi Li, Tianqing Fang, Hongming Zhang, Haitao Mi, Dong Yu, and Zhicheng Dou. 2025. Unigist: Towards general and hardware-aligned sequence-level long context compression. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Geoffrey B. Duggan and Stephen J. Payne. 2011. Skim reading by satisficing: evidence from eye tracking. In *CHI*, pages 1141–1150. ACM.

Alexander R. Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir R. Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *ACL (1)*, pages 1074–1084. Association for Computational Linguistics.

Yixiong Fang, Tianran Sun, Yuling Shi, and Xiaodong Gu. 2025. Attentionrag: Attention-guided context pruning in retrieval-augmented generation. *CoRR*, abs/2503.10720.

Tao Ge, Jing Hu, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. 2024. In-context autoencoder for context compression in a large language model. In *ICLR*. OpenReview.net.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing A multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *COLING*, pages 6609–6625. International Committee on Computational Linguistics.

Taeho Hwang, Sukmin Cho, Soyeong Jeong, Hoyun Song, SeungYoon Han, and Jong C. Park. 2025a. EXIT: context-aware extractive compression for enhancing retrieval-augmented generation. In *ACL (Findings)*, pages 4895–4924. Association for Computational Linguistics.

Taeho Hwang, Sukmin Cho, Soyeong Jeong, Hoyun Song, SeungYoon Han, and Jong C. Park. 2025b. EXIT: Context-aware extractive compression for enhancing retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 4895–4924, Vienna, Austria. Association for Computational Linguistics.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023. Llmlingua: Compressing prompts for accelerated inference of large language models. In *EMNLP*, pages 13358–13376. Association for Computational Linguistics.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2024. Longllmlingua: Accelerating and enhancing llms in long context scenarios via prompt compression. In *ACL (1)*, pages 1658–1677. Association for Computational Linguistics.

Tomás Kociský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The narrativeqa reading comprehension challenge. *Trans. Assoc. Comput. Linguistics*, 6:317–328.

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 *NeurIPS*.

Yucheng Li, Bo Dong, Frank Guerin, and Chenghua Lin. 2023. Compressing context to enhance inference efficiency of large language models. In *EMNLP*, pages 6342–6353. Association for Computational Linguistics.

Zongqian Li, Yixuan Su, and Nigel Collier. 2025. 500xcompressor: Generalized prompt compression for large language models. In *ACL (1)*, pages 25081–25091. Association for Computational Linguistics.

Huanxuan Liao, Wen Hu, Yao Xu, Shizhu He, Jun Zhao, and Kang Liu. 2025. Beyond hard and soft: Hybrid context compression for balancing local and global information retention. *CoRR*, abs/2505.15774.

Barys Liskavets, Maxim Ushakov, Shuvendu Roy, Mark Klibanov, Ali Etemad, and Shane K. Luke. 2025. Prompt compression with context-aware sentence encoding for fast and improved LLM inference. In *AAAI*, pages 24595–24604. AAAI Press.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Trans. Assoc. Comput. Linguistics*, 12:157–173.

Jesse Mu, Xiang Li, and Noah D. Goodman. 2023. Learning to compress prompts with gist tokens. In *NeurIPS*.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Dongmei Zhang. 2024. Llmlingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. In *ACL (Findings)*, pages 963–981. Association for Computational Linguistics.

Aleksandar Petrov, Mark Sandler, Andrey Zhmoginov, Nolan Miller, and Max Vladymyrov. 2025. Long context in-context compression by getting to the gist of gisting. *CoRR*, abs/2504.08934.

David Rau, Shuai Wang, Hervé Déjean, and Stéphane Clinchant. 2024. Context embeddings for efficient answer generation in RAG. *CoRR*, abs/2407.09252.

Claude E. Shannon. 1948. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27(3):379–423.

Sijun Tan, Xiuyu Li, Shishir G. Patil, Ziyang Wu, Tianjun Zhang, Kurt Keutzer, Joseph Gonzalez, and Raluca A. Popa. 2024. Lloco: Learning long contexts offline. In *EMNLP*, pages 17605–17621. Association for Computational Linguistics.

Jiwei Tang, Jin Xu, Tingwei Lu, Zhicheng Zhang, Yiming Zhao, Lin Hai, and Hai-Tao Zheng. 2025a. Perception compressor: A training-free prompt compression framework in long context scenarios. In *NAACL (Findings)*, pages 4093–4108. Association for Computational Linguistics.

Jiwei Tang, Zhicheng Zhang, Shunlong Wu, Jingheng Ye, Lichen Bai, Zitai Wang, Tingwei Lu, Jiaqi Chen, Lin Hai, Hai-Tao Zheng, and Hong-Gee Kim. 2025b. GMSA: enhancing context compression via group merging and layer semantic alignment. *CoRR*, abs/2505.12215.

Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, and 1 others. 2025. Kimi k2: Open agentic intelligence. *Preprint*, arXiv:2507.20534.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*, pages 5998–6008.

Maryanne Wolf. 2018. *Reader, come home: The reading brain in a digital world*. Harper, an imprint of HarperCollinsPublishers.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2024. RECOMP: improving retrieval-augmented lms with context compression and selective augmentation. In *ICLR*. OpenReview.net.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, and 1 others. 2024. Qwen2 technical report. *CoRR*, abs/2407.10671.

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 *EMNLP*, pages 2369–2380. Association for Computational Linguistics.

Xubing Ye, Yukang Gan, Xiaoke Huang, Yixiao Ge, and Yansong Tang. 2025. Voco-llama: Towards vision compression with large language models. In *CVPR*, pages 29836–29846. Computer Vision Foundation / IEEE.

Chanwoong Yoon, Taewhoo Lee, Hyeon Hwang, Minbyul Jeong, and Jaewoo Kang. 2024. Compact: Compressing retrieved documents actively for question answering. In *EMNLP*, pages 21424–21439. Association for Computational Linguistics.

Peitian Zhang, Zheng Liu, Shitao Xiao, Ninglu Shao, Qiwei Ye, and Zhicheng Dou. 2025. Long context compression with activation beacon. In *ICLR*. OpenReview.net.

Yunlong Zhao, Haoran Wu, and Bo Xu. 2025. Leveraging attention to effectively compress prompts for long-context llms. In *AAAI*, pages 26048–26056. AAAI Press.

A Impact of Segment Size

As shown in Table 4, we conduct experiments to investigate the impact of varying segment sizes on RAM's performance. Under a 4x compression constraint with Qwen3-4B-Instruct as the backbone, we observe that the model consistently achieves strong and stable results across all three QA benchmarks. The low standard deviations demonstrate minimal performance variance across different segment granularities, highlighting the robustness and reliability of RAM in practical deployment scenarios.

Table 4: Experimental results with varying segment sizes on three QA benchmarks using Qwen3-4B-Instruct as backbone under 4x compression constraint.

Seg. Size	2WikiMQA		HotpotQA		NarrativeQA	
	EM	F1	EM	F1	EM	F1
50	50.39	56.47	46.37	59.28	19.45	31.92
100	48.84	54.15	47.27	59.64	18.98	31.94
200	47.52	52.90	46.06	58.41	19.74	31.32
Std.	1.44	1.81	0.63	0.63	0.38	0.35

B Prompt Template of Annotation

We employ a prompt template (see Figure 5) to instruct Qwen3-235B-A22B-Instruct to label each segment in NarrativeQA as positive if it is helpful for generating the ground-truth answer, and negative otherwise.

C Dataset Details

NaturalQuestions This dataset consists of real queries issued to the Google search engine, paired with entire Wikipedia pages. It requires models to identify both a long answer (typically a paragraph) and a short answer (one or more entities or a boolean) within the document. The corpus contains 307,373 training examples, 7,830 development examples, and 7,842 test examples. It is widely utilized for evaluating open-domain question answering and machine reading comprehension under realistic information-seeking scenarios. *We select the processed version (Liu et al., 2024; Cao et al., 2024) where each question has 20 related documents and only one of them contains the correct answer. The processed version has 2,655 test samples.*

HotpotQA HotpotQA is a large-scale dataset designed for multi-hop reasoning over Wikipedia articles. Questions are constructed such that the answer can only be found by performing reasoning across multiple documents. It features 112,779 training samples and 7,405 validation samples. A distinguishing factor of this dataset is the requirement for models to provide supporting facts (sentences) to explain the reasoning process, which enhances the explainability of the question-answering systems. *We use the validation set to evaluate model's performance.*

2WikiMQA This dataset focuses on multi-hop question answering with structured evidence, specifically utilizing Wikipedia and Wikidata. It aims to minimize the presence of "reasoning shortcuts" by using templates to generate complex questions that require synthesizing information from up to four documents. The dataset includes 167,454 training instances, 12,576 validation instances, and 12,576 test instances. It includes four types of reasoning: compositional, inference, comparison, and bridge. *We use the test set to evaluate model performance.*

NarrativeQA NarrativeQA is designed to test deep understanding of entire stories rather than local surface-level matching. It contains questions based on complete books and movie scripts. Unlike datasets that rely on short snippets, this benchmark requires models to capture long-range dependencies. The collection includes 1,572 documents divided into 1,102 for training, 115 for validation, and 355 for testing. These documents correspond to 46,765 total question and answer pairs.*We use the test set to evaluate the model's performance, filtering out test samples longer than 32K.*

MultiNews As a large-scale multi-document summarization dataset, Multi-News consists of news articles and human-written summaries. Each sample contains a summary paired with multiple source articles from various news sites. The dataset provides 44,972 training examples, 5,622 validation examples, and 5,622 test examples. It is a standard benchmark for evaluating the ability of models to consolidate redundant information and resolve conflicting details across diverse sources. *We use the test set to evaluate model performance.*

D Ethic Statement

This paper introduces RAM, an efficient and interpretable context compression framework inspired

Prompt Template

Given the following question and segment, determine whether the segment contains sufficient information to answer the question. Respond with only "Yes" or "No".

Question: {question}

Segment: {context}

Answer:

Figure 5: Prompt template of annotation used by Qwen3-235B-A22B-Instruct.

by human reading behavior. It combines parallelizable close reading and query-guided skimming to retain key information while reducing redundancy. The data and models used in this work are sourced from publicly available benchmarks and open-source platforms under appropriate licenses. While our method may influence how long-context LLMs are deployed, it does not introduce new ethical risks beyond those already present in existing context compression approaches. Thus, no additional ethical concerns require specific attention.

E Language Model Usage Statement

In drafting this paper, we use a large language model to assist with academic writing. Specifically, we use it to improve wording, organization, and overall readability, including edits to the description of our methods and to the exposition of mathematical derivations. The scientific contributions of this work, including its key ideas, experimental setup, and reported results, are conceived, executed, and verified by the authors.