DIFFERENTIAL ATTENTION FOR CONTEXT SELECTION

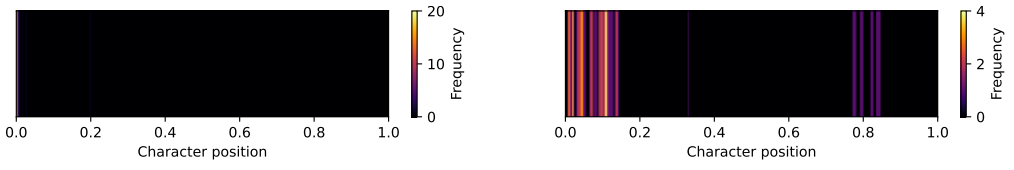
Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models with long context windows can answer questions over full-length documents, but using the entire document is costly, slow, and not always more accurate. This paper studies long-context evidence selection: given a long document, a query, and a strict token budget, the goal is to expose only the most useful evidence to a downstream generator. Standard retrieval-augmented generation (RAG) methods retrieve semantically similar chunks using strong embedding models, but they can miss evidence that is sparse, structurally distributed, or weakly matched to the query. Prompt-compression methods, such as LongLLMLingua and CPC, reduce context in another way, but often do not explicitly model where query-specific evidence appears. We propose *Selective Attention-Guided Extraction* (SAGE), a training-free, plug-and-play context reduction framework. SAGE uses a lightweight frozen LLM during prefilling, without decoding or fine-tuning, to turn query-to-document attention into a document-level relevance heatmap. It then applies differential attention with a contrast query to suppress query-irrelevant background signals and isolate question-specific evidence. Finally, SAGE selects coherent high-scoring spans under the token budget and passes only the reduced context to a downstream generation LLM. Across QuALITY-hard, NarrativeQA, QASPER, and AIT-QA, SAGE improves the accuracy–token trade-off over strong embedding-based RAG baselines using Qwen3-Embedding and ColBERT, as well as prompt-compression baselines such as LongLLMLingua and CPC. On QuALITY-hard, SAGE achieves a top-4 leaderboard result using only a 30% context budget, showing that carefully selected attention-guided evidence can outperform much larger retrieved or compressed contexts.

1 INTRODUCTION

Long-context large language models (LLMs) are increasingly used to answer questions over complex documents, including scientific papers, policy documents, legal notices, reports, and semi-structured records (Liu et al., 2025b;a; Lin et al., 2025; Ma et al., 2024; Qian et al., 2024; Weng et al., 2024; Zhang et al., 2024a; Corallo & Papotti, 2024). Although recent models can ingest entire documents end-to-end (Liu et al., 2025b), more context is not always better. Long inputs often mix useful evidence with irrelevant text, which can distract the model and degrade answer accuracy (Jin et al., 2024; Liu et al., 2024b). We argue that long-context reasoning is not only a capacity problem,



(a) Who are the authors of the paper

(b) What is the main contribution of the paper

Figure 1: Heatmaps of ground-truth answer distributions in the Paper dataset. Author queries in a cluster near the beginning of the document, while contribution queries in b are distributed more broadly.

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.

but also an *evidence selection* problem: in our experiments, a carefully selected 15% of a document can perform similarly to a dense retriever using a top embedding model with 40% of the document.

Evidence selection is also important for efficiency. Full-document processing increases pay-per-token API cost, wastes compute in self-hosted settings, and becomes difficult when documents exceed a model's context window. These costs motivate *context reduction*: selecting a compact subset of the input that preserves answer quality while bounding LLM cost.

Retrieval-Augmented Generation (RAG) commonly reduces context before generation, but its reliance on embedding similarity can miss answer-bearing evidence. For example, for the query "Who are the authors of this paper?", RAG may retrieve the reference section because author names appear in citations, while the correct answer is usually in the header. This suggests that useful evidence depends not only on semantic similarity, but also on document structure and position.

This motivates estimating where relevant evidence is likely to appear. Prior work (Lin et al., 2025) and our observations show that answer evidence often follows positional patterns, such as author information appearing near the beginning of a paper. Validation-based positional heatmaps exploit these patterns. Given a small set of documents for the same query type, a strong LLM can test candidate regions, generate answers, and match them back to source spans. Aggregating these positions across documents yields a heatmap that can guide extraction on new documents.

However, validation-based heatmaps provide only a rough estimate of where evidence appears. They require labeled examples or a strong LLM to locate answer spans beforehand, and capture average answer positions rather than evidence for a specific query in a specific document. As shown in Fig. 1, author queries cluster near the beginning of the paper, while contribution queries are more broadly distributed. For reasoning-heavy questions, evidence may span multiple sections, making a fixed positional region insufficient. These limitations motivate a more adaptive heatmap. During prefilling, Transformer-based LLMs compute attention from query tokens to preceding context tokens, yielding a fine-grained, query-conditioned map over the document before generation. We use this attention as a training-free relevance heatmap, requiring only the prefill pass and avoiding the dominant cost of decoding (Agrawal et al., 2025).

This leads to our first research question: *Can the internal attention of a frozen LLM serve as a training-free relevance signal for long-context evidence selection?* We answer this with **S**elective **A**ttention-**G**uided **E**xtraction (SAGE), which uses a lightweight frozen LLM to compute prefill-only attention, aggregates the resulting scores into coherent text units, and selects high-scoring evidence under a user-specified token budget. The selected original text is then passed to a downstream LLM for generation.

Raw attention, however, may assign high scores to broadly salient but query-irrelevant tokens, such as titles, section headings, or repeated phrases. This raises a second question: *How can we remove query-irrelevant noise while preserving query-specific evidence?* SAGE addresses this with *differential attention*. Inspired by contrastive approaches that separate shared background noise from task-specific signals (Ye et al., 2024; Lim et al., 2025; Ge et al., 2025), we subtract the attention induced by a contrast query from that of the target query, suppressing shared signals and sharpening the relevance heatmap.

We evaluate SAGE across long-context, document, and semi-structured QA settings. On QuALITY-hard (Pang et al., 2022), SAGE ranks **4th** on the public leaderboard (Pang et al.) while using only a **30%** context budget. On QASPER (Dasigi et al., 2021) and NarrativeQA-derived(Kočiský et al., 2018), it consistently outperforms strong embedding-based RAG and context-reduction baselines. On *Paper* and *Notice* (Lin et al., 2025), we further study token budgets, selector model size, and differential-attention variants.

To summarize, our contributions are:

- We formulate long-context QA as budgeted, query-specific evidence selection and propose SAGE, a training-free framework that uses prefill-only attention from a frozen lightweight LLM to select original text for a separate downstream generator.
- We introduce differential attention with a contrast query to suppress query-irrelevant background signals. We further show that attention-derived relevance rankings are query-specific and remain consistent across different selector models.

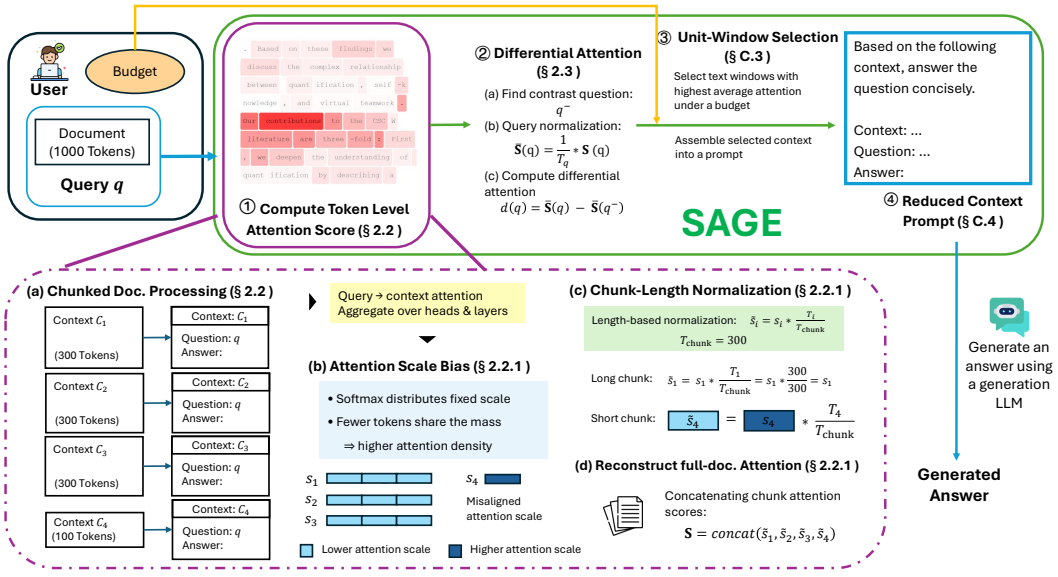


Figure 2: **Overview of the SAGE pipeline.** ① *Attention computation* utilizes chunked processing and length normalization to form a document-level score. ② *Differential attention* refines relevance by filtering structural noise. ③ *Unit-window selection* extracts top text spans under the budget. ④ Spans are assembled into a *reduced-context prompt* for final answer generation.

- We show that SAGE improves the accuracy–context trade-off across long-context QA benchmarks. On QuALITY-hard, it reaches 63.1% accuracy using about 30% of the context, within 2.25 percentage points of full-context generation and achieving a top-4 result on the QuALITY-hard leaderboard.
- We study when and how attention-guided selection remains effective. Fine-grained selection operates below the size of a single retrieval chunk, smaller selectors retain much of the performance of 8B models, and row-level selection on AIT-QA nearly matches full-table accuracy while using about half of the rows.

2 Methodology

We now formalize long-context evidence selection and present SAGE. Given a document, query, and token budget, SAGE first constructs a query-conditioned attention heatmap, then denoises it with differential attention, and finally selects compact evidence for downstream generation.

Fig. 2 summarizes the full pipeline. This section focuses on its two central components: constructing the attention heatmap and denoising it with differential attention.

2.1 Problem Definition

Given a long document $\mathbf{D} = (x_1, \ldots, x_{T_D})$, a query $q = (q_1, \ldots, q_{T_q})$, and a budget ratio $b \in (0, 1]$, we aim to answer q from only a compact subset of $\mathbf{D}$. Let M be a frozen lightweight LLM used as an evidence filter, and let G be the downstream generative LLM. SAGE selects a budgeted context

$$\mathbf{D}_b = \mathrm{SAGE}(\mathbf{D}, q, b; M), \qquad |\mathbf{D}_b| \le B = \lceil bT_D \rceil,$$

and the final answer is generated from the reduced input:

$$G(\mathbf{D}_b, q) = G\big(\mathrm{SAGE}(\mathbf{D}, q, b; M), q\big).$$

Here, M uses attention only to rank document regions, while G generates the answer from the selected original text. The challenge is therefore to construct $\mathbf{D}_b$.

2.2 ATTENTION HEATMAPS

SAGE first uses the frozen selector model M to assign each document token a query-conditioned relevance score. As shown in Stage ① of Fig. 2, these scores are computed from query-to-context attention and form an attention heatmap over the document.

Since the full document may exceed the context window of M, SAGE computes the heatmap chunk by chunk. We split the document into contiguous chunks $\mathbf{D} = \{C_1, \ldots, C_n\}$, where $C_i = (c_{i,1}, \ldots, c_{i,T_i})$ contains at most T_{chunk} tokens and preserves the original document order. For each chunk, we construct

$$\texttt{Context:}\ \ C_i\ \ \texttt{Question:}\ \ q\ \ \texttt{Answer:}$$

and run only the prefilling pass of M. Under causal self-attention, query tokens can attend to earlier context tokens; therefore, their attention to C_i gives a query-conditioned signal of which tokens are relevant. When multiple queries share the same document, the chunk-prefix KV cache can be reused across queries, avoiding repeated encoding of the document tokens; implementation details are provided in Appendix C.2.

For the t-th token in chunk C_i, the raw attention score is

$$s_{i,t}(q) = \sum_{k=1}^{T_q} \sum_{\ell \in \mathcal{L}} \sum_{h \in \mathcal{H}} A_{\ell,h}(q_k, c_{i,t}), \tag{1}$$

where $A_{\ell,h}(q_k, c_{i,t})$ is the attention assigned by query token q_k to context token $c_{i,t}$ at layer ℓ and head h of M. Tokens with larger $s_{i,t}(q)$ are treated as stronger evidence candidates for the query. We sum attention across all layers and heads because the highest-scoring heads and layers can vary across inputs. Appendix B further analyzes the reliability of using attention as a relevance signal.

2.2.1 CHUNK NORMALIZATION

Chunking makes attention extraction feasible for M, but introduces two issues. First, shorter chunks distribute attention over fewer tokens, which can inflate their per-token scores. This often affects the final chunk. We therefore apply a length-based correction to reduce score-scale differences across chunks: $\tilde{s}_{i,t}(q) = s_{i,t}(q) \cdot T_i / T_{\text{chunk}}$, where T_i is the length of chunk C_i. This reduces the scale of shorter chunks while preserving token rankings within each chunk. After normalization, we concatenate chunk scores in document order to form the full-document heatmap, with one score per token:

$$\mathbf{S}(q) = \text{concat}\left(\tilde{\mathbf{s}}_1(q), \tilde{\mathbf{s}}_2(q), \ldots, \tilde{\mathbf{s}}_n(q)\right). \tag{2}$$

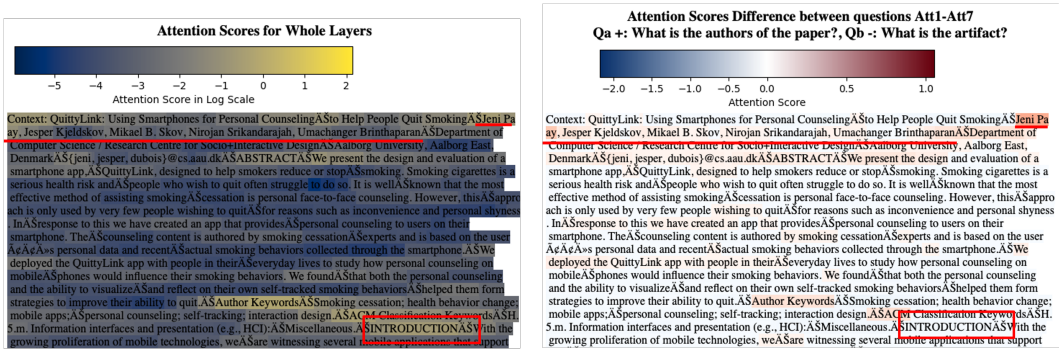
Second, relevant spans may be split across chunk boundaries. We therefore lightly smooth the reconstructed scores, allowing high-attention tokens to provide partial support to nearby tokens. This reflects that evidence usually appears as a short span rather than as an isolated token. Details are in Appendix C.1. For simplicity, we continue to denote the smoothed document-level heatmap by $\mathbf{S}(q)$.

2.3 DIFFERENTIAL ATTENTION

After normalization and smoothing, $\mathbf{S}(q)$ provides a query-conditioned relevance map over the document. However, because it is derived from the raw attention of M, high scores do not always indicate answer evidence. Fig. 3 shows this issue for the query "Who are the authors of the paper?". Although the author information receives strong attention, the structural token "Introduction" receives an even higher score, potentially wasting the context budget.

We therefore compare the target query with a contrast query over the same document. As shown in Fig. 3b, subtracting the contrast heatmap suppresses attention shared by both queries, such as section headers and formatting patterns, while preserving signals more specific to the target query. The resulting differential attention provides a cleaner signal for selecting the context passed to G.

Since $\mathbf{S}(\cdot)$ sums attention over query tokens, queries of different lengths can have different score scales. We therefore normalize each heatmap by query length: $\bar{\mathbf{S}}(q) = \frac{1}{T_q}\mathbf{S}(q)$ and $\bar{\mathbf{S}}(q^-) =$



(a) Raw attention for the target query (b) Differential attention with a contrast query

Figure 3: **Effect of differential attention.** For the target query "Who are the authors of the paper?", raw attention from M highlights structural tokens such as "Introduction". Using "What is the artifact?" as the contrast query suppresses shared background attention and makes author evidence more visible. Positive values favor the target query; negative values favor the contrast query.

$\frac{1}{T_{q^-}}\mathbf{S}(q^-)$, where T_q and T_{q^-} are the numbers of tokens in the two queries. The differential attention score is then

$$\mathbf{d}(q) = \bar{\mathbf{S}}(q) - \bar{\mathbf{S}}(q^-). \tag{3}$$

We use $\mathbf{d}(q)$ as the denoised heatmap for context selection.

2.3.1 Choosing the Contrast Query

The contrast query controls what background signal is removed. A good q^- should share document-level noise with the target query, while focusing on different answer evidence. If q^- is too similar to q, subtraction may remove useful evidence.

We consider two contrast-query strategies and include raw attention as a baseline:

- **Raw attention.** We use $\mathbf{S}(q)$ directly, without differential subtraction.
- **Fixed contrast query.** We use the same contrast prompt for all examples: *"Please repeat the context."* This estimates generic document salience without requiring other questions for the same document.
- **Farthest contrast query.** When multiple questions are available for the same document, we choose as q^- the question whose embedding is farthest from the target query embedding. This encourages different answer evidence while keeping the same document background. We further examine how the choice of contrast query affects performance in Appendix F.1.1.

Farthest contrast strategy requires at least one additional query for the same document. When no such question is available, SAGE can always use the Fixed query without additional training.

After obtaining the denoised heatmap $\mathbf{d}(q)$, SAGE selects coherent text spans under the token budget using a unit-window selection strategy. We rank contiguous windows by their average score, select them under the remaining token budget, and merge overlapping windows into coherent spans. The selected original text forms $\mathbf{D}_b$ and is assembled in document order into a reduced-context prompt for G. We defer the full selection algorithm to Appendix C.3.

3 Evaluation

We evaluate SAGE across long-context QA, document QA, and semi-structured table QA. The evaluation is organized around six observations: (1) **SAGE improves accuracy with moderate runtime overhead**; (2) **long-context QA is an evidence-selection problem, not only a capacity problem**; (3) **differential attention filters query-irrelevant noise**; (4) **SAGE's framework can be extended to semi-structured data**; (5) **fine-grained attention selection remains useful below the size of a single retrieval chunk**; and (6) **larger attention models provide cleaner evidence-selection signals**.

3.1 EXPERIMENTAL SETUP

Datasets. We use QuALITY-hard (Pang et al., 2022), QASPER (Dasigi et al., 2021), and a NarrativeQA-derived long-context benchmark (Kočiský et al., 2018) for the main comparison. We additionally use Paper and Notice from ZenDB (Lin et al., 2025) for document-level analyses, and AIT-QA (Katsis et al., 2022) for table QA. The separate FINCH comparison uses three LongBench (Bai et al., 2024) tasks. Dataset construction and task differences are detailed in Appendix D.1.

Baselines. We compare with dense RAG using Qwen3-Embedding, UAE-Large, and Octen-Embedding, the late-interaction retriever ColBERT (Khattab & Zaharia, 2020), and the prompt-compression methods LongLLMLingua (Jiang et al., 2024b) and CPC (Liskavets et al., 2025). We also compare with AttentionRAG (Fang et al., 2025) at similar measured token usage and report FINCH (Corallo & Papotti, 2024) separately because it compresses KV-cache states rather than selecting text. The default budget selection is described in Appendix D.2.

Models and evaluation. Our text-selection experiments use *Gemini-2.5-Flash-Lite* as the generator G with the same decoding setup. Unless otherwise stated, SAGE uses frozen *Qwen3-8B* as the selector M, Farthest contrast selected with *all-MiniLM-L6-v2*, and *GPT-4o-mini* for correctness evaluation. Local model computation uses NVIDIA A100 GPUs. For Paper and Notice, we additionally measure answer similarity using *Qwen3-Embedding-0.6B*. Model and evaluation details are in Appendices D.3 and D.4.

3.2 RESULTS

Table 1: **Main results at budget** 0.3**.** Accuracy and average runtime in seconds per query. Retrieval depths approximate the target budget. NarrativeQA-derived uses the setting described in Sec. 3.1. Best accuracy on each benchmark is bolded.

Method	QuALITY-hard		QASPER		NarrativeQA-derived	
	Acc.	Time	Acc.	Time	Acc.	Time
RAG (UAE-Large)	0.536	0.83	0.545	5.30	0.621	0.48
RAG (Qwen3-Emb.)	0.562	1.00	0.595	0.85	0.687	0.72
RAG (Octen-Emb.)	0.575	0.78	0.610	1.04	0.713	0.51
ColBERT	0.544	0.96	0.579	1.03	0.780	0.93
LongLLMLingua	0.446	7.01	0.461	3.22	0.489	1.42
CPC	0.589	2.70	0.561	2.29	0.852	1.07
SAGE (Qwen3-8B)	**0.631**	2.37	**0.614**	3.59	**0.880**	0.85

Observation 1: SAGE improves accuracy with moderate runtime overhead. Table 1 reports the main results under a 0.3 context budget. Across QuALITY-hard, QASPER, and the NarrativeQA-derived setting, SAGE achieves the highest accuracy among retrieval-based and prompt-compression baselines. The gain is especially clear on QuALITY-hard, where SAGE reaches 0.631 accuracy, compared with 0.589 for CPC and 0.575 for the strongest dense RAG baseline. SAGE also improves over the best baseline on QASPER (0.614 vs. 0.610) and NarrativeQA-derived (0.880 vs. 0.852), showing that attention-guided selection generalizes across different long-context QA regimes.

The runtime remains within a practical range. Although SAGE is not always the fastest method, it is not the most expensive overall. For QuALITY-hard, SAGE is both faster and more accurate than LongLLMLingua and CPC. It also runs faster than LongLLMLingua, CPC, and ColBERT on the NarrativeQA-derived benchmark. Although SAGE has higher runtime than most baselines on QASPER, it achieves the best accuracy and remains faster than RAG with UAE-Large. These results suggest that the accuracy gains come with moderate runtime overhead, rather than an impractically expensive selection pipeline.

Table 2: **Budget sweep on QuALITY-hard.** For SAGE and prompt-compression methods, columns denote token budgets. For retrieval methods, columns denote the top-k retrieved chunks chosen to approximate the corresponding token budget. Best results in each column are bolded.

Method	0.10 / k=3	0.15 / k=5	0.30 / k=7	0.40 / k=10
LongLLMLingua	0.368	0.404	0.446	0.466
CPC	0.464	0.514	0.589	0.597
RAG (Qwen3-Emb.)	0.460	0.534	0.562	0.585
RAG (UAE-Large)	0.444	0.497	0.536	0.584
RAG (Octen-Emb.)	0.489	0.545	0.575	0.596
SAGE (Llama-3.2-1B)	0.454	0.517	0.597	0.631
SAGE (Qwen3-8B)	**0.520**	**0.570**	**0.631**	**0.641**

We further compare SAGE with AttentionRAG at nearly matched context usage, as shown in Table 10. SAGE retains 10.07% of the original context, while AttentionRAG retains 10.25%. Under this matched context usage, SAGE achieves 52.02% accuracy compared with 39.62% for AttentionRAG, an improvement of 12.40 percentage points.

FINCH scores are reported from the original paper using Mistral-7B-Instruct-v0.2, whereas SAGE uses Gemini-2.5-Flash-Lite. Because the generators and budget definitions differ, this comparison provides reference results rather than a controlled comparison of compression methods. As shown in Table 11, SAGE reports higher F1 on all three tasks under their respective settings: 54.24 versus 44.13 for FINCH on MultiFieldQA-EN, 18.09 versus 17.60 on NarrativeQA, and 33.65 versus 23.19 on QASPER. The corresponding F1 differences are 10.11, 0.49, and 10.46 points, respectively. Detailed results are provided in Appendix E.

Observation 2: long-context QA is an evidence-selection problem, not only a capacity problem. A central motivation of SAGE is that long-context QA is often limited by whether the prompt contains the right evidence, not simply by how many tokens the generator can read. Table 2 supports this view on QuALITY-hard. We compare SAGE across several context budgets and attention models against RAG baselines with strong retrieval embeddings. Across the tested budget range, SAGE achieves a better accuracy–token trade-off, reaching higher accuracy while passing less context to the generator. In particular, SAGE with Qwen3-8B achieves the highest accuracy at every tested budget. As the budget increases, the performance gaps narrow and the accuracy gains from adding more context become smaller. Moreover, at a 15% target budget, SAGE reaches 57.0% accuracy, which is similar to dense RAG baselines using approximately 40% of the context. Rather than simply adding more retrieved chunks, SAGE uses query-conditioned attention to identify the regions most likely to support the answer.

Table 3 further compares this setting with full-context generation, where the entire original document is passed to the same Gemini-2.5-Flash-Lite generator under the same generation and evaluation setup. Full context reaches 65.35% accuracy, while SAGE reaches 63.10% using only 30.09% of the original context on average. This reduces the document context passed to the generator by 69.91% with an accuracy gap of only 2.25 percentage points, further showing diminishing returns from providing additional context once the relevant evidence has been retained.

Table 3: **Comparison with full-context generation on QuALITY-hard.** Both settings use all evaluation examples and the same generation and evaluation setup.

Method	Accuracy	Avg. Token Pct.
Full context	**65.35%**	100.00%
SAGE ($b = 0.30$)	63.10%	30.09%

SAGE also ranks **4th** on QuALITY-hard using this 30% budget with Qwen3-8B, without fine-tuning the selector or training a task-specific retriever. Together, these results show that the main challenge in long-context QA is not only fitting more text into the context window, but selecting the evidence that actually matters.

Table 4: **QA accuracy on Paper under different attention strategies.** All variants use Qwen3-8B for selection, Gemini-2.5-Flash-Lite for generation, and GPT-4o-mini for evaluation. Raw uses attention without contrastive subtraction. Fixed uses a fixed contrast query, and Farthest uses a semantically distant question from the same document. Best accuracy in each column is bolded.

Attention strategy	Accuracy (%) at target budget b			
	0.005	0.10	0.15	0.40
Raw	10.64	35.82	39.39	62.08
Fixed	28.21	**57.65**	**64.10**	**73.04**
Farthest	**32.71**	54.93	60.30	71.95

Observation 3: differential attention filters query-irrelevant noise. Raw attention provides a useful relevance signal, but it can also assign high scores to generally salient regions that are not specific to the current query. To test whether differential attention removes this background signal, we compare Raw attention with Fixed and Farthest contrast on the Paper dataset.

As shown in Table 4, both differential-attention variants outperform Raw at every tested budget. The improvement is particularly large at small and moderate budgets. For example, at $b = 0.10$, accuracy increases from 35.82% with Raw to 57.65% with Fixed and 54.93% with Farthest. Neither contrast strategy dominates across budgets. Farthest performs best at the smallest budget, while Fixed performs best at the other tested budgets. This also supports Fixed as a useful alternative when no additional question is available for the document. These results support differential attention as a denoising step that produces a more query-specific evidence signal. Additional analyses of answer similarity, contrast-query selection, and the components used to normalize the attention signal are provided in Appendix F.

Observation 4: SAGE's framework extends to semi-structured data. SAGE is not tied to plain-text span selection. The token attention scores can be aggregated over other units, such as table rows. On AIT-QA (Katsis et al., 2022), we serialize each table, compute query-aware attention, aggregate token scores into row scores, and select the top rows with their table headers. As shown in Table 5, SAGE reaches 0.87 accuracy using only 51% of rows, nearly matching the full-table baseline at 0.88. This shows that attention-guided selection can extend to semi-structured inputs without task-specific training.

Table 5: **AIT-QA table selection results.** SAGE nearly matches full-table accuracy while using about half of the table rows.

Method	**Accuracy**	**Avg. Row Usage**
Full Table	0.88	100%
SAGE	0.87	51%

Observation 5: Fine-grained selection remains useful below a retrieval chunk. With the fixed chunk sizes used by our RAG baselines, retrieving one chunk exposes the entire chunk to the generator, even when the answer depends on only a small part of it. SAGE instead selects coherent token-level spans under an explicit budget, allowing it to retain only the relevant portion of a chunk or combine evidence across chunks.

This finer granularity is useful when evidence is short and localized. On Notice, including the notice-date question, SAGE remains effective at context sizes below the first available RAG setting and generally matches or outperforms RAG at comparable usage. Thus, context reduction depends not only on retrieving the right chunk, but also on selecting the specific evidence within and across chunks. Complete results and semantic-similarity analysis are in Appendix E.3.

Observation 6: Larger selectors provide modest gains, but small selectors remain competitive. We vary only the frozen selector while keeping the generator and evaluation setup fixed (Table 6). Accuracy increases consistently with selector size in both model families. For Llama, accuracy rises from 59.7% with the 1B selector to 62.3% with the 8B selector, a gain of only 2.6 percentage points. For Qwen, scaling from 0.6B to 8B improves accuracy from 58.4% to 63.1%, a 4.7-point gain. The intermediate models perform nearly identically across families, with Llama-3.2-3B at 61.1% and Qwen3-4B at 61.2%. These results show that larger selectors provide consistent but moderate gains, while much smaller models retain most of the downstream QA performance.

Table 6: **Selector-model comparison on QuALITY-hard.** Accuracy (%) at $b = 0.30$ with Farthest contrast. Only the selector changes; the Gemini-2.5-Flash-Lite generator, evaluation prompt, and GPT-4o-mini judge remain fixed. Larger selectors achieve higher accuracy within both tested families, while smaller selectors retain much of that performance. The highest accuracy within each family is bolded.

Llama selector	Accuracy (%)	Qwen selector	Accuracy (%)
Llama-3.2-1B	59.7	Qwen3-0.6B	58.4
Llama-3.2-3B	61.1	Qwen3-4B	61.2
Llama-3.1-8B	**62.3**	Qwen3-8B	**63.1**

4 Related Work

Prior work has studied several ways to reduce the cost of long-context LLM inference. Additional discussion is provided in Appendix A.

Prompt and context compression. Prompt compression aims to shorten the input while preserving the information needed for generation (Li et al., 2025). Some methods reduce the effective input through learned representations, soft prompts, or optimized prompt components, including CC (Wingate et al., 2022), GIST (Mu et al., 2024), ICAE (Ge et al., 2023b), xRAG (Cheng et al., 2024), and MIPRO (Opsahl-Ong et al., 2024). These approaches can be compact, but they often require training, are difficult to inspect, and may distort the original text (Li et al., 2025; Liskavets et al., 2025). Other methods perform hard compression by pruning or selecting discrete tokens, such as PoWER-BERT (Goyal et al., 2020), TR-BERT (Ye et al., 2021), SpAtten (Wang et al., 2021), LTP (Kim et al., 2022), LLMLingua (Jiang et al., 2023), and LongLLMLingua (Jiang et al., 2024a). More recent context reduction methods preserve larger text units, such as CPC (Liskavets et al., 2025) and ZenDB (Lin et al., 2025). Attn-GS (Zeng et al., 2026) also uses LLM attention as a relevance signal. It aggregates token-level attention into sentence scores, marks relevant sentences using a relative threshold, and uses a separate model to rewrite the selected interaction history into a reusable profile. In contrast, SAGE does not train a compressor, generate a summary, or rewrite the context. It uses prefill-only attention from a frozen lightweight LLM to select original text spans under an explicit budget.

Differential and contrastive attention. Attention structure has been widely studied for extraction and long-context reasoning (Wu et al., 2024; Zhang et al., 2024b). Differential and contrastive attention mechanisms have also been used to separate task-specific signal from shared background patterns. Early work introduced Differential Attention Networks in vision by contrasting supporting and opposing exemplars (Patro & Namboodiri, 2018). More recent LLM methods, such as Differential Transformer (Ye et al., 2024) and Grouped Differential Attention (Lim et al., 2025), modify the attention computation itself to cancel common-mode noise. Contrastive signals have also been used to focus multimodal models on task-relevant visual regions (Ge et al., 2025) and to suppress generic text during decoding (Li et al., 2023). SAGE is different: it does not modify the Transformer or intervene during decoding. Instead, it applies differential attention as a training-free, prefilling-stage filter to denoise attention heatmaps before context selection.

5 Conclusion

We presented SAGE, a training-free, plug-and-play framework for long-context evidence selection. SAGE uses prefill-only attention from a lightweight frozen LLM to build query-specific relevance heatmaps, and refines them with differential attention to remove query-irrelevant background signals. Across long-context QA, document QA, and semi-structured table QA benchmarks, SAGE improves the accuracy–token trade-off over strong retrieval and prompt-compression baselines. Its reusable document representations also make it well suited to repeated-query settings.

While selection quality depends on the attention signal and may be affected by redundant evidence, the selected original text remains directly inspectable, and future work can further improve contrast-query construction and evidence selection.

6 STATEMENTS

AI USE STATEMENT

The research ideas and proposed method were developed by the authors. The authors wrote the initial manuscript and implementation. We used generative AI tools to refine the language and presentation of the manuscript, assist with code development, and provide feedback on the consistency and interpretation of the reported experimental results. The authors reviewed the AI-assisted revisions and examined the implementation. We take responsibility for the final manuscript, code, experimental claims, and conclusions.

Separately, the use of LLMs for evidence selection, answer generation, and correctness evaluation is part of our experimental pipeline, as described in Section 3.1 and Appendix D.

ETHICS STATEMENT

This work studies evidence selection for document and table question answering using existing benchmarks. Although SAGE preserves selected original text for inspection, it may omit necessary context or retain distracting evidence, leading to incomplete or unsupported answers. The selected context should therefore remain available for verification. In applications involving consequential decisions, generated answers should be checked against the original source rather than treated as reliable solely because they are based on selected evidence.

REPRODUCIBILITY STATEMENT

Section 2 describes attention-heatmap construction, normalization, and differential attention. Appendix C provides implementation details, including smoothing, KV-cache reuse, the budgeted unit-window selection algorithm, and the reduced-context prompt. Section 3.1 and Appendix D describe the datasets, baselines, budget alignment, model choices, and evaluation metrics. Appendix E provides additional baseline comparisons and per-question results.

REFERENCES

Amey Agrawal, Nitin Kedia, Anmol Agarwal, Jayashree Mohan, Nipun Kwatra, Souvik Kundu, Ramachandran Ramjee, and Alexey Tumanov. On evaluating performance of llm inference serving systems. *arXiv preprint arXiv:2507.09019*, 2025.

Vaswani Ashish. Attention is all you need. *Advances in neural information processing systems*, 2017.

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, 2024.

Guoxuan Chen, Han Shi, Jiawei Li, Yihang Gao, Xiaozhe Ren, Yimeng Chen, Xin Jiang, Zhenguo Li, Weiyang Liu, and Chao Huang. Sepllm: Accelerate large language models by compressing one segment into one separator. *arXiv preprint arXiv:2412.12094*, 2024a.

Yilong Chen, Guoxia Wang, Junyuan Shang, Shiyao Cui, Zhenyu Zhang, Tingwen Liu, Shuohuan Wang, Yu Sun, Dianhai Yu, and Hua Wu. Nacl: A general and effective kv cache eviction framework for llm at inference time. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7913–7926, 2024b.

Xin Cheng, Xun Wang, Xingxing Zhang, Tao Ge, Si-Qing Chen, Furu Wei, Huishuai Zhang, and Dongyan Zhao. xrag: Extreme context compression for retrieval-augmented generation with one token. *Advances in Neural Information Processing Systems*, 37:109487–109516, 2024.

Giulio Corallo and Paolo Papotti. FINCH: Prompt-guided key-value cache compression for large language models. *Transactions of the Association for Computational Linguistics*, 12:1517–1532, 2024. doi: 10.1162/tacl_a_00716. URL https://aclanthology.org/2024.tacl-1.83/.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A Smith, and Matt Gardner. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4599–4610, 2021.

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

Suyu Ge, Yunan Zhang, Liyuan Liu, Minjia Zhang, Jiawei Han, and Jianfeng Gao. Model tells you what to discard: Adaptive kv cache compression for llms. *arXiv preprint arXiv:2310.01801*, 2023a.

Tao Ge, Jing Hu, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. In-context autoencoder for context compression in a large language model. *arXiv preprint arXiv:2307.06945*, 2023b.

Yuyao Ge, Shenghua Liu, Yiwei Wang, Lingrui Mei, Baolong Bi, Xuanshan Zhou, Jiayu Yao, Jiafeng Guo, and Xueqi Cheng. Focusing by contrastive attention: Enhancing vlms' visual reasoning. *arXiv preprint arXiv:2509.06461*, 2025.

Saurabh Goyal, Anamitra Roy Choudhury, Saurabh Raje, Venkatesan Chakaravarthy, Yogish Sabharwal, and Ashish Verma. Power-bert: Accelerating bert inference via progressive word-vector elimination. In *International Conference on Machine Learning*, pp. 3690–3699. PMLR, 2020.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Llmlingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 13358–13376, 2023.

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, 2024a.

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, 2024b. URL https://arxiv.org/abs/2310.06839.

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

Bowen Jin, Jinsung Yoon, Jiawei Han, and Sercan O Arik. Long-context llms meet rag: Overcoming challenges for long inputs in rag. *arXiv preprint arXiv:2410.05983*, 2024.

Jiajie Jin, Xiaoxi Li, Guanting Dong, Yuyao Zhang, Yutao Zhu, Yongkang Wu, Zhonghua Li, Ye Qi, and Zhicheng Dou. Hierarchical document refinement for long-context retrieval-augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3502–3520, 2025.

Hyukkyu Kang, Injung Kim, and Wook-Shin Han. Trial: Token relations and importance aware late-interaction for accurate text retrieval. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 16875–16888, 2025.

Yannis Katsis, Saneem Chemmengath, Vishwajeet Kumar, Samarth Bharadwaj, Mustafa Canim, Michael Glass, Alfio Gliozzo, Feifei Pan, Jaydeep Sen, Karthik Sankaranarayanan, et al. Ait-qa: Question answering dataset over complex tables in the airline industry. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track*, pp. 305–314, 2022.

Omar Khattab and Matei Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pp. 39–48, 2020.

Sehoon Kim, Sheng Shen, David Thorsley, Amir Gholami, Woosuk Kwon, Joseph Hassoun, and Kurt Keutzer. Learned token pruning for transformers. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 784–794, 2022.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328, 2018. doi: 10.1162/tacl_a_00023. URL https://aclanthology.org/Q18-1023.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori B Hashimoto, Luke Zettlemoyer, and Mike Lewis. Contrastive decoding: Open-ended text generation as optimization. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 12286–12312, 2023.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. Snapkv: Llm knows what you are looking for before generation. *Advances in Neural Information Processing Systems*, 37:22947–22970, 2024.

Zongqian Li, Yinhong Liu, Yixuan Su, and Nigel Collier. Prompt compression for large language models: A survey. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 7182–7195, 2025.

Junghwan Lim, Sungmin Lee, Dongseok Kim, Wai Ting Cheung, Beomgyu Kim, Taehwan Kim, Haesol Lee, Junhyeok Lee, Dongpin Oh, and Eunhwan Park. Grouped differential attention. *arXiv preprint arXiv:2510.06949*, 2025.

Yiming Lin, Madelon Hulsebos, Ruiying Ma, Shreya Shankar, Sepanta Zeighami, Aditya G Parameswaran, and Eugene Wu. Querying templatized document collections with large language models. In *2025 IEEE 41st International Conference on Data Engineering (ICDE)*, pp. 2422–2435. IEEE, 2025.

Barys Liskavets, Maxim Ushakov, Shuvendu Roy, Mark Klibanov, Ali Etemad, and Shane K Luke. Prompt compression with context-aware sentence encoding for fast and improved llm inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 24595–24604, 2025.

Chunwei Liu, Matthew Russo, Michael J Cafarella, Lei Cao, Peter Baile Chen, Zui Chen, Michael J Franklin, Tim Kraska, Samuel Madden, Rana Shahout, et al. Palimpzest: Optimizing ai-powered analytics with declarative query processing. In *CIDR*, 2025a.

Di Liu, Meng Chen, Baotong Lu, Huiqiang Jiang, Zhenhua Han, Qianxi Zhang, Qi Chen, Chengruidong Zhang, Bailu Ding, Kai Zhang, et al. Retrievalattention: Accelerating long-context llm inference via vector retrieval. *arXiv preprint arXiv:2409.10516*, 2024a.

Jiaheng Liu, Dawei Zhu, Zhiqi Bai, Yancheng He, Huanxuan Liao, Haoran Que, Zekun Wang, Chenchen Zhang, Ge Zhang, Jiebin Zhang, et al. A comprehensive survey on long context language modeling, 2025. *URL https://arxiv. org/abs/2503.17407*, 2025b.

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

Yubo Ma, Yuhang Zang, Liangyu Chen, Meiqi Chen, Yizhu Jiao, Xinze Li, Xinyuan Lu, Ziyu Liu, Yan Ma, Xiaoyi Dong, et al. Mmlongbench-doc: Benchmarking long-context document understanding with visualizations. *Advances in Neural Information Processing Systems*, 37:95963–96010, 2024.

Jesse Mu, Xiang Li, and Noah Goodman. Learning to compress prompts with gist tokens. *Advances in Neural Information Processing Systems*, 36, 2024.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

Krista Opsahl-Ong, Michael J Ryan, Josh Purtell, David Broman, Christopher Potts, Matei Zaharia, and Omar Khattab. Optimizing instructions and demonstrations for multi-stage language model programs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 9340–9366, 2024.

Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, Jana Thompson, He He, and Samuel Bowman. QuALITY Leaderboard — nyu-mll.github.io. `https://nyu-mll.github.io/quality/`. [Accessed 01-03-2026].

Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, Jana Thompson, He He, and Samuel Bowman. QuALITY: Question answering with long input texts, yes! In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5336–5358, Seattle, United States, July 2022. Association for Computational Linguistics. URL `https://aclanthology.org/2022.naacl-main.391`.

Badri Patro and Vinay P Namboodiri. Differential attention for visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7680–7688, 2018.

Rui Qian, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Shuangrui Ding, Dahua Lin, and Jiaqi Wang. Streaming long video understanding with large language models. *Advances in Neural Information Processing Systems*, 37:119336–119360, 2024.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. Colbertv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3715–3734, 2022.

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

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 5797–5808, 2019.

Hanrui Wang, Zhekai Zhang, and Song Han. Spatten: Efficient sparse attention architecture with cascade token and head pruning. In *2021 IEEE international symposium on high-performance computer architecture (HPCA)*, pp. 97–110. IEEE, 2021.

Yuetian Weng, Mingfei Han, Haoyu He, Xiaojun Chang, and Bohan Zhuang. Longvlm: Efficient long video understanding via large language models. In *European Conference on Computer Vision*, pp. 453–470. Springer, 2024.

David Wingate, Mohammad Shoeybi, and Taylor Sorensen. Prompt compression and contrastive conditioning for controllability and toxicity reduction in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 5621–5634, 2022.

Wenhao Wu, Yizhong Wang, Guangxuan Xiao, Hao Peng, and Yao Fu. Retrieval head mechanistically explains long-context factuality. *arXiv preprint arXiv:2404.15574*, 2024.

Jingbo Yang, Bairu Hou, Wei Wei, Yujia Bao, and Shiyu Chang. Kvlink: Accelerating large language models via efficient kv cache reuse. *arXiv preprint arXiv:2502.16002*, 2025.

Deming Ye, Yankai Lin, Yufei Huang, and Maosong Sun. Tr-bert: Dynamic token reduction for accelerating bert inference. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5798–5809, 2021.

Tianzhu Ye, Li Dong, Yuqing Xia, Yutao Sun, Yi Zhu, Gao Huang, and Furu Wei. Differential transformer. *arXiv preprint arXiv:2410.05258*, 2024.

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

Shenglai Zeng, Tianqi Zheng, Chuan Tian, Dante Everaert, Yau-Shian Wang, Yupin Huang, Michael J Morais, Rohit Patki, Jinjin Tian, Xinnan Dai, et al. Attn-gs: Attention-guided context compression for efficient personalized llms. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 35925–35946, 2026.

Peiyuan Zhang, Kaichen Zhang, Bo Li, Guangtao Zeng, Jingkang Yang, Yuanhan Zhang, Ziyue Wang, Haoran Tan, Chunyuan Li, and Ziwei Liu. Long context transfer from language to vision. *arXiv preprint arXiv:2406.16852*, 2024a.

Zhenyu Zhang, Runjin Chen, Shiwei Liu, Zhewei Yao, Olatunji Ruwase, Beidi Chen, Xiaoxia Wu, and Zhangyang Wang. Found in the middle: How language models use long contexts better via plug-and-play positional encoding. *Advances in Neural Information Processing Systems*, 37: 60755–60775, 2024b.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, et al. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems*, 36, 2024c.

Xiabin Zhou, Wenbin Wang, Minyan Zeng, Jiaxian Guo, Xuebo Liu, Li Shen, Min Zhang, and Liang Ding. Dynamickv: Task-aware adaptive kv cache compression for long context llms. *arXiv preprint arXiv:2412.14838*, 2024.

A Additional Related Work

This section provides a fuller discussion of related work on long-context RAG, prompt compression, differential attention, and KV-cache compression.

A.1 Long-Context RAG

Retrieval-Augmented Generation (RAG) reduces input length by conditioning generation on retrieved evidence rather than the full document. RAG systems include several retrieval designs. Early systems often used dense single-vector encoders, while late interaction retrievers such as ColBERT (Khattab & Zaharia, 2020), ColBERTv2 (Santhanam et al., 2022), and TRIAL (Kang et al., 2025) improve localization by keeping token-level matching information. As long-context LLMs became more common, retrieval systems also moved toward larger and more coherent retrieval units, as in LongRAG (Jiang et al., 2024c), or hierarchical evidence organization, as in RAPTOR (Sarthi et al., 2024). More recently, AttentionRAG (Fang et al., 2025) uses model attention to prune retrieved contexts, reformulating the query into a next-token prediction form so that query-context attention can identify useful context tokens. Together, these methods reduce the amount of text passed to the generator, but they still operate within a retrieval pipeline and depend on retrieved candidates or retrieved contexts as the starting point.

However, retrieving more text does not necessarily improve answer quality. Jin et al. (2024) show that adding more retrieved passages can distract the reader model with "hard negatives": passages that are similar to the query in embedding space but irrelevant or misleading for the answer. This motivates refinement systems such as LongRefiner (Jin et al., 2025) and OP-RAG (Yu et al., 2024). SAGE is complementary to these systems. RAG can first retrieve candidate documents or chunks, while SAGE can further select the most useful evidence within those candidates at a finer granularity.

A.2 KV Cache Compression

A complementary line of work reduces memory and latency during decoding by optimizing the KV cache without changing the input prompt. FastGen (Ge et al., 2023a) and SnapKV (Li et al., 2024) profile attention patterns to compress caches for heavy-hitting tokens. FINCH (Corallo & Papotti, 2024) performs prompt-guided KV-cache compression during prefilling by selecting the most relevant key-value pairs for a given prompt, enabling long-input processing without fine-tuning. For dynamic cache management, H_2O (Zhang et al., 2024c) formulates KV eviction as a submodular optimization problem, while NACL (Chen et al., 2024b) improves eviction robustness through proxy-token estimation. Beyond eviction, RetrievalAttention (Liu et al., 2024a) manages cache access through approximate nearest-neighbor search, and SepLLM (Chen et al., 2024a) uses separator tokens as compact anchors. Other optimizations include dynamic cache budget allocation across layers (Zhou et al., 2024) and concatenating precomputed caches across repeated contexts, as in KVLink (Yang et al., 2025).

These methods are highly useful for reducing memory and latency during long-context inference. However, most of them still assume that the long context has already been ingested by the model, especially during the prefilling stage. SAGE instead reduces the input context before generation. This can lower both prefilling and decoding costs, while also giving users explicit control over the amount of text exposed to the downstream generator.

B Reliability of Attention as a Relevance Signal

This section provides additional justification for using attention as a relevance signal in SAGE. We first explain its mechanistic motivation, then test whether different selector models produce consistent query-specific rankings, and finally examine why we aggregate attention across all heads and layers.

B.1 Mechanistic Motivation for Attention

Query-to-context attention provides a model-native signal for evidence ranking. SAGE measures how strongly each query token attends to each context token. Attention forms token representations as weighted combinations of value vectors based on query–key compatibility (Ashish, 2017). Highly attended context tokens therefore contribute more strongly to the query-conditioned representations, making attention an internal information-routing signal rather than an external similarity score.

Raw attention can also capture query-independent patterns, such as positional bias, syntax, or generally salient content. SAGE reduces this noise using differential attention. The target and contrast queries are applied to the same context with the same selector, and their query-length-normalized heatmaps are subtracted according to Eq. 3. This suppresses shared patterns while preserving query-specific differences.

Importantly, SAGE does not treat attention as a complete causal explanation of the final answer, nor assume that every highly attended token is answer-bearing. Instead, attention is used as a relevance-ranking signal. We next test whether this signal remains consistent across different selector models.

B.2 Cross-Model Consistency of Attention Rankings

Comparison protocol. We study whether different selector models produce consistent query-specific relevance rankings over the same document. For each pair of selectors, we compare their agreement for the *same* query with their agreement for *different* queries over the same document. Because the document is fixed, higher agreement for the same query would indicate that the shared ranking contains query-specific information rather than only generic document structure.

We evaluate six selectors on 220 valid QuALITY-hard documents: Qwen3-0.6B, Qwen3-4B, Qwen3-8B, Llama-3.2-1B, Llama-3.2-3B, and Llama-3.1-8B. Documents without a valid farthest-question pair are excluded. For each document, the first question is the anchor $q^{(1)}$, and its recorded farthest-query id identifies $q^{(2)}$. The two queries therefore represent different information needs over the same document. We use superscripts to distinguish these complete queries from the token indices q_k used in the main text.

To isolate the effect of the target query, both $q^{(1)}$ and $q^{(2)}$ use the same fixed contrast query, q^-_{fixed}: "Please repeat the context." Let M_m denote selector m, and let $\mathbf{S}^{\text{raw}}_m(q)$ denote its raw attention score vector arranged in document order. Following the query normalization in the main method, we define $\bar{\mathbf{S}}^{\text{raw}}_m(q) = \mathbf{S}^{\text{raw}}_m(q)/T_q$. We then compute

$$\mathbf{d}^{\text{raw}}_m(q) = \bar{\mathbf{S}}^{\text{raw}}_m(q) - \bar{\mathbf{S}}^{\text{raw}}_m(q^-_{\text{fixed}}).$$

Here, raw indicates that these scores are taken directly from the model before the remaining document-level processing used by SAGE. Using the same contrast query across selectors and target queries keeps the background reference fixed, so the comparison reflects how the target query changes the relevance signal.

Region-level rankings. We compare rankings rather than raw attention values for two reasons. First, different models may produce attention scores on different scales. Second, Qwen and Llama use different tokenizers, making token-level alignment of their attention vectors unreliable. We therefore compare their attention rankings as an approximate cross-model alignment. More importantly, SAGE uses attention to rank document regions for selection, rather than relying on the absolute score values.

We therefore divide each $\mathbf{d}^{\text{raw}}_m(q)$ into 100 consecutive relative-position regions and average the scores within each region. Let $\mathbf{r}_m(q)$ denote the resulting region-level score vector. We use Spearman correlation $\rho \in [-1, 1]$ to measure how similarly two selectors rank these regions.

For selectors M_a and M_b, we compute

$$\rho_{\text{same}} = \frac{\rho(\mathbf{r}_a(q^{(1)}), \mathbf{r}_b(q^{(1)})) + \rho(\mathbf{r}_a(q^{(2)}), \mathbf{r}_b(q^{(2)}))}{2}, \tag{4}$$

$$\rho_{\text{different}} = \frac{\rho(\mathbf{r}_a(q^{(1)}), \mathbf{r}_b(q^{(2)})) + \rho(\mathbf{r}_a(q^{(2)}), \mathbf{r}_b(q^{(1)}))}{2}, \tag{5}$$

$$\Delta_{\text{Spearman}} = \rho_{\text{same}} - \rho_{\text{different}}. \tag{6}$$

Table 7: Cross-model Spearman correlations over 220 documents for matched (Same) and different (Different) queries over the same document, grouped by model-family relation. Δ may differ from Same $-$ Different in the last digit due to rounding.

Model A	Model B	Same	Different	Δ
Within Qwen				
Qwen3-0.6B	Qwen3-4B	0.780	0.506	+0.274
Qwen3-0.6B	Qwen3-8B	0.788	0.506	+0.282
Qwen3-4B	Qwen3-8B	0.832	0.503	+0.329
Within Llama				
Llama-3.2-1B	Llama-3.2-3B	0.819	0.554	+0.265
Llama-3.2-1B	Llama-3.1-8B	0.742	0.475	+0.267
Llama-3.2-3B	Llama-3.1-8B	0.833	0.505	+0.329
Cross family				
Qwen3-0.6B	Llama-3.2-1B	0.745	0.504	+0.241
Qwen3-0.6B	Llama-3.2-3B	0.721	0.465	+0.257
Qwen3-0.6B	Llama-3.1-8B	0.704	0.431	+0.272
Qwen3-4B	Llama-3.2-1B	0.683	0.448	+0.235
Qwen3-4B	Llama-3.2-3B	0.700	0.424	+0.275
Qwen3-4B	Llama-3.1-8B	0.726	0.415	+0.312
Qwen3-8B	Llama-3.2-1B	0.704	0.461	+0.244
Qwen3-8B	Llama-3.2-3B	0.729	0.439	+0.291
Qwen3-8B	Llama-3.1-8B	0.755	0.425	+0.331

Table 8: Mean cross-model ranking agreement by model-family relation, averaged over the model pairs in each group.

Relation	Same	Different	Δ
Within Qwen	0.800	0.505	+0.295
Within Llama	0.798	0.511	+0.287
Cross family	0.719	0.446	+0.273

A positive $\Delta_{\mathrm{Spearman}}$ indicates that the two selectors agree more strongly when ranking regions for the same query than for different queries over the same document.

Results and interpretation. Across all 15 model pairs, the mean same-query correlation is 0.751, compared with 0.471 for different queries, yielding an average $\Delta_{\mathrm{Spearman}}$ of $+0.280$. The pattern holds both within and across model families. Within Qwen and Llama, the mean same-query correlations are 0.800 and 0.798, respectively, while cross-family pairs still reach 0.719. Every model pair has a positive $\Delta_{\mathrm{Spearman}}$, and the same ordering holds across the evaluated document–model pairs.

Because both queries come from the same document, this comparison controls for document structure and position. The consistently higher same-query correlations therefore indicate that the shared rankings contain a query-dependent component. These results support the consistency of attention-derived relevance rankings across the tested selectors. They do not imply that every highly ranked region contains answer evidence or that different selectors produce identical rankings.

B.3 Why We Aggregate Across All Attention Heads

Prior work shows that attention heads can capture different functions, including positional, syntactic, or induction patterns (Voita et al., 2019; Olsson et al., 2022). Such specialization does not establish that a fixed subset is sufficient for query-specific evidence selection.

In an exploratory analysis, the highest-ranked layer–head pairs varied across three contexts for the same question (Table 9). Only $(19, 1)$ appears in more than one context, and no pair appears in all three. This suggests that a fixed subset may omit useful signals for some inputs.

Table 9: Highest-ranked layer–head pairs for the same question across three contexts. Each entry is a (layer, head) pair from the exploratory analysis.

Rank	Context 1	Context 2	Context 3
1	$(13, 27)$	$(26, 7)$	$(15, 29)$
2	$(19, 1)$	$(23, 5)$	$(19, 1)$
3	$(19, 19)$	$(30, 21)$	$(13, 26)$

We therefore aggregate across all layers, heads, and query tokens rather than selecting a preset subset. This avoids an additional head-selection procedure and retains signals distributed across the model. Preliminary comparisons with preset layer or head subsets were less stable and did not consistently improve performance. These observations motivate the design but do not prove that full aggregation is optimal. Differential attention then suppresses patterns shared with the contrast query, including background signals retained by full aggregation.

C Method Details

C.1 Chunk Smoothing

Since many documents exceed the context window of the selector model, SAGE computes attention heatmaps chunk by chunk. We tokenize the full document using the selector model's tokenizer and split it into contiguous, non-overlapping chunks of at most T_{chunk} tokens. In our implementation, $T_{\text{chunk}} = 2000$. The chunks preserve the original document order, but because we do not use overlap, a relevant passage may be split across two adjacent chunks.

After computing token-level attention scores for each chunk, we apply chunk-length normalization and concatenate the normalized chunk heatmaps back into the original document order. This gives a single document-level heatmap

$$\mathbf{S}(q) = (S_1(q), \ldots, S_T(q)).$$

However, because adjacent chunks are processed independently, the stitched heatmap may contain artificial discontinuities around chunk boundaries. This can be problematic when a coherent evidence span is cut in the middle by the chunking process.

To reduce this boundary artifact, we apply a light smoothing step after the chunk heatmaps have been normalized and concatenated:

$$\mathbf{S}^{\text{smooth}}(q) = \text{Smooth}(\mathbf{S}(q)).$$

Importantly, smoothing is applied to the reconstructed document-level heatmap, not to each chunk independently. This allows local information to pass across chunk boundaries and makes the heatmap less sensitive to the exact chunk split. Query-length normalization is applied later when constructing differential-attention heatmaps, so smoothing is kept separate from query normalization.

We consider two standard one-dimensional smoothing methods. Gaussian smoothing replaces each token score with a distance-weighted local average of its neighbors, which helps reduce high-frequency noise. Savitzky–Golay smoothing fits a low-order polynomial inside a local sliding window, which can smooth local fluctuations while better preserving sharp peaks. In the main experiments, we use Savitzky–Golay smoothing with window length 11 and polynomial order 3. We clip negative values introduced by smoothing before constructing differential attention. Negative differential-attention scores are retained unchanged and passed to unit-window selection.

Fig. 4 compares several Savitzky–Golay settings on the paper dataset using Llama-3.2-1B with the farthest-query differential-attention variant at budget 0.4. The complementary CDF curves are nearly overlapping across window lengths and polynomial orders, indicating that the final answer quality is not highly sensitive to the exact smoothing hyperparameters in this range. This suggests that smoothing mainly acts as a lightweight stabilization step for stitched heatmaps, rather than as a major source of performance gain. We therefore use a moderate default setting throughout the main experiments.

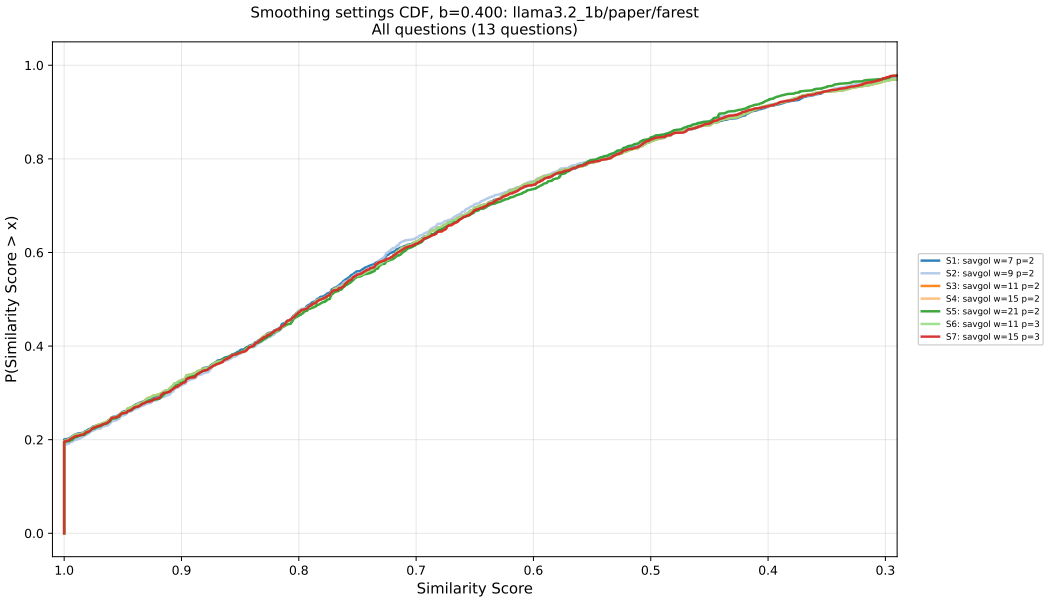


Figure 4: Complementary CDF of answer similarity scores under different Savitzky–Golay smoothing settings on the paper dataset. We use Llama-3.2-1B as the selector model, the farthest-query differential-attention variant, and budget $b = 0.4$. The curves are nearly overlapping across window lengths and polynomial orders, suggesting that performance is not highly sensitive to the exact smoothing hyperparameters within the tested range.

C.2 KV-CACHE REUSE FOR EFFICIENT ATTENTION COMPUTATION.

Computing attention independently for every chunk–query pair would require repeatedly encoding the same chunk content for different queries. To eliminate this redundancy, we reuse the model's key–value (KV) cache.

We decompose the chunk–query input into two parts. The *prefix* is the context section, while the *suffix* is the query section. The query text is inserted after the `Question:` marker and varies across queries.

For each chunk C_i, we encode the prefix once and store the resulting KV states. These cached states represent the hidden representations of the chunk tokens and can be reused for multiple queries. We index cache entries using a hash of the chunk token IDs and maintain the cache per document. When switching to a new document, all cached states are cleared to prevent cross-document reuse.

When processing a query q for chunk C_i, we retrieve the cached KV states and run the model only on the query suffix, supplying the cached states as past key values. During this step, each query token attends to the cached context tokens, allowing the model to produce the same attention scores as if the full prefix–suffix sequence had been processed jointly. We then aggregate these attention scores to compute $s_{i,t}$ without re-encoding the chunk.

This cache-based execution preserves the causal computation, up to numerical differences. Each chunk is encoded once and can be reused across many queries, reducing both computation and memory overhead.

C.3 BUDGETED UNIT-WINDOW SELECTION

After obtaining the denoised token-level heatmap $\mathbf{d}(q)$, SAGE must convert token scores into a budgeted evidence subset $\mathbf{D}_b$. Although $\mathbf{d}(q)$ assigns one score to each document token, selecting isolated high-scoring tokens would produce a fragmented context and remove the local information needed by the generator. We therefore select contiguous token windows and merge overlapping windows into coherent spans. Algorithm 1 gives the full procedure.

Algorithm 1 Budgeted Unit-Window Selection

Require: Tokenized document $\mathbf{D}$, heatmap $\mathbf{h}$, budget ratio b
Ensure: Budgeted evidence subset $\mathbf{D}_b$

$T_D \leftarrow |\mathbf{D}|, \quad B \leftarrow \lceil bT_D \rceil, \quad w \leftarrow 0.02$
if $T_D = 0$ **then**
  return $\emptyset$
end if
if $b < w$ **then**
  Set window length $L \leftarrow B$
  Select the length-L window with the highest average heatmap score
  return the selected window as $\mathbf{D}_b$
end if
Set window length $L \leftarrow \lceil wT_D \rceil$
Score every length-L sliding window by its average heatmap score
Sort windows from highest score to lowest score
$\mathcal{S} \leftarrow \emptyset$ ▷ selected token positions
for each window W in sorted order **do**
  $r \leftarrow B - |\mathcal{S}|$ ▷ remaining token budget
  if $r = 0$ **then**
    break
  end if
  $\mathcal{U} \leftarrow W \setminus \mathcal{S}$ ▷ new token positions
  if $|\mathcal{U}| \leq r$ **then**
    $\mathcal{S} \leftarrow \mathcal{S} \cup W$
  else
    Add the first r positions of $\mathcal{U}$ in document order to $\mathcal{S}$
  end if
end for
Merge adjacent selected positions in $\mathcal{S}$ into contiguous spans
Sort the spans by document position
Concatenate the spans to form $\mathbf{D}_b$
return $\mathbf{D}_b$

Let $\mathbf{d}(q) = (d_1, \ldots, d_{T_D})$, where d_t is the denoised relevance score of the t-th document token. Given a budget ratio $b \in (0, 1]$, the total selection budget is

$$B = \lceil bT_D \rceil.$$

We use a base window ratio $w = 0.02$. For normal budgets, $b \geq w$, we use a fixed window length

$$L = \lceil wT_D \rceil,$$

so that SAGE can select multiple coherent evidence regions across the document. For very small budgets, $b < w$, using the base window would already exceed the budget. In this case, Algorithm 1 sets

$$L = B$$

and selects only the highest-scoring window. This special case avoids splitting a tiny budget into many short fragments.

For a window starting at token position i, we compute its average score as

$$u(i) = \frac{1}{L} \sum_{t=i}^{i+L-1} d_t, \qquad 1 \leq i \leq T_D - L + 1.$$

When $b < w$, Algorithm 1 selects the single window with the largest $u(i)$. When $b \geq w$, it ranks all sliding windows by $u(i)$ and greedily adds high-scoring windows until the budget B is reached. For each candidate window, only token positions that have not already been selected count toward the remaining budget. If all newly introduced tokens fit, the full window is added. Otherwise, only the first r newly introduced positions in document order, where r is the remaining budget, are retained. Adjacent selected positions are then merged into contiguous spans. The final selected spans are sorted by their original document positions and concatenated to form $\mathbf{D}_b$.

C.4 Reduced-Context Prompt

After unit-window selection, SAGE obtains a budgeted evidence subset $\mathbf{D}_b$ from the original document. We then place $\mathbf{D}_b$ and the query q into a fixed prompt template for the downstream generator G:

```
Based on the following context, answer the question
concisely.

Context: {selected context}

Question: {query}

Answer:
```

All methods use the same generator prompt format whenever possible, so the comparison focuses on the quality of the selected context rather than prompt differences. The selected spans are concatenated in their original document order before being inserted into the context field. This preserves the document flow while exposing only the evidence chosen under the budget.

D Additional Experimental Details

D.1 Dataset Details

We provide additional details about the datasets used in our evaluation. The main experiments use QuALITY-hard, QASPER, and a NarrativeQA-derived long-context benchmark. We additionally use Paper and Notice for differential-attention and budget-sensitivity analysis, and AIT-QA for semi-structured table QA.

QuALITY-hard. QuALITY (Pang et al., 2022) is a long-context multiple-choice QA benchmark built from English fiction and non-fiction passages. Each question has four candidate answers. We evaluate on 1,065 examples from QuALITY-hard, whose questions annotators could not reliably answer under time-constrained search but could answer with sufficient reading time. This subset tests whether context selection preserves evidence that may be distributed across a passage.

QASPER. QASPER (Dasigi et al., 2021) contains questions about NLP papers, with evidence drawn from different sections. Its answer formats include free-form, yes/no, and extractive answers. All methods receive the paper text and use the same downstream generator, allowing us to evaluate context selection beyond multiple-choice QA.

NarrativeQA-derived benchmark. NarrativeQA (Kočiský et al., 2018) is based on books and movie scripts. We use `illuin-conteb/narrative-qa`[1], which retains the original questions but represents source documents as collections of chunks. We group chunks by source document and concatenate them before context selection. Because this may not exactly reconstruct the original documents, we report it as a NarrativeQA-derived long-context setting rather than a direct reproduction of full-document NarrativeQA.

Paper. Paper comes from ZenDB (Lin et al., 2025) and pairs scientific papers with multiple questions per document. These include localized metadata questions about authors, titles, and publication information, as well as open-ended questions about the main contribution. This mixture supports analysis of both compact and distributed evidence. Multiple questions per document also allow same-document contrast-query selection.

Notice. Notice also comes from ZenDB (Lin et al., 2025) and contains regulatory notices with formal, repetitive language. Questions target fields such as the notice date, company name, violation items, and address information. We use it to study fine-grained selection when the required evidence is much shorter than a retrieval chunk and may be surrounded by repeated fields.

[1] `https://huggingface.co/datasets/illuin-conteb/narrative-qa`

AIT-QA. AIT-QA (Katsis et al., 2022) contains tables from airline SEC filings with complex headers and domain-specific content. We serialize each table into a JSON-style representation, compute query-aware token attention, and aggregate token scores into row scores. We then select the top-ranked rows and retain their table headers for generation. Table 5 reports the average fraction of rows retained, rather than a token-retention budget.

LongBench tasks for the FINCH comparison. We separately evaluate MultiFieldQA-EN, NarrativeQA, and QASPER from LongBench (Bai et al., 2024). This comparison uses F1 and is distinct from the main benchmark setting, including the reconstructed NarrativeQA-derived contexts and LLM-judge accuracy. We use Fixed contrast because the LongBench inputs used here do not provide document identifiers for constructing a same-document question pool.

D.2 Budget Selection

For SAGE and prompt-compression methods, the budget is specified as a fraction of the original context length. For example, budget 0.30 means that the method keeps about 30% of the original context. Retrieval methods are instead controlled by the number of retrieved chunks. To align them with token budgets, we split documents into roughly 200-token chunks with 50-token overlap, estimate the average token usage at different retrieval depths, and choose the top-k value whose average length is closest to the target budget.

Choice of main budget. Table 2 reports a budget sweep on QuALITY-hard. Very small budgets, such as 0.10 and 0.15, test whether a method can find highly concentrated evidence, but they can be too restrictive when answers require multiple evidence pieces. Larger budgets, such as 0.40, provide more context, but the gains often flatten. For example, CPC improves only from 0.589 at budget 0.30 to 0.597 at budget 0.40, while RAG with Octen-Embedding improves from 0.575 to 0.596.

We use budget 0.30 for the main benchmark table because it gives a balanced comparison point: the generator receives only about one third of the original context, but the budget is still large enough for difficult long-context QA. At this budget, SAGE with Qwen3-8B reaches 0.631, compared with 0.589 for CPC and 0.575 for the strongest dense RAG baseline. The sweep also shows that SAGE remains strong across budgets, and that a stronger selector improves evidence selection: Qwen3-8B consistently outperforms Llama-3.2-1B, especially under tighter budgets.

D.3 Model Selection

To enable a fair and controlled comparison, we separate the models used for attention computation, retrieval, answer generation, and evaluation. All methods share the same downstream generation and evaluation setup whenever possible, so differences in final performance mainly reflect differences in the selected context.

Attention models. For SAGE, we evaluate frozen Llama-3.2-1B, Llama-3.2-3B, Llama-3.1-8B, Qwen3-0.6B, Qwen3-4B, and Qwen3-8B as selector models. These models are used only to estimate query-to-context attention and select evidence under the token budget. They do not generate the final answers. This design lets us study how attention-model capacity affects evidence selection while keeping the answer generator fixed. Unless stated otherwise, SAGE uses the farthest-query differential-attention variant when constructing the reduced context.

Answer generation and LLM-based evaluation. All methods, including SAGE, dense RAG, ColBERT, LongLLMLingua, and CPC, use *Gemini-2.5-Flash-Lite* for answer generation. We fix the generator and decoding configuration across methods to isolate the effect of context selection. For LLM-based evaluation, we use *GPT-4o-mini* as the judge across datasets. The judge is given dataset-specific grading instructions and outputs a binary correctness decision.

Embedding models. Embedding models are used for three different purposes. First, dense RAG baselines use strong embedding models, including Qwen3-Embedding, UAE-Large, and Octen-Embedding, for chunk retrieval. Second, for farthest-query selection in differential attention, we use *all-MiniLM-L6-v2*. This model is lightweight and sufficient for estimating relative query distance

within the same document. Third, for semantic-similarity evaluation, we use *Qwen3-Embedding-0.6B*. This model provides stronger semantic representations while remaining efficient enough for large-scale evaluation.

D.4 Evaluation Metrics

We evaluate answer quality using two complementary metrics: *LLM-based judgment* and *semantic similarity*. LLM-based judgment gives a task-aware binary correctness label, while semantic similarity gives a softer measure of how closely the generated answer matches the reference meaning.

LLM-based correctness. We report accuracy as the fraction of answers judged correct. For QuALITY-hard, the judge checks whether the final answer selects the gold option. For QASPER, NarrativeQA-derived, Paper, Notice, and AIT-QA, dataset-specific instructions allow equivalent wording while rejecting missing required information, unsupported content, or answers to a different question. All methods within a dataset use the same evaluation prompt and judge, except where a separate experimental protocol is explicitly identified.

Manual inspection and judging prompt. We manually inspected GPT-4o-mini's decisions and refined the evaluation prompts based on observed errors. The QuALITY-hard prompt below instructs the judge to identify the final selected option rather than intermediate analysis. We map output `10` to correct and `0` to incorrect when computing accuracy.

```
Question: {question}

Model's answer: {prediction}

Ground truth: {ground_truth}

Does the model's answer match the ground truth?

The model may express its choice as a letter, boxed or bold text,
a statement such as "The answer is A," or by restating the option.

Identify the model's final answer and ignore intermediate analysis.

Output 10 if the model selected option {ground_truth};
otherwise output 0.
```

Semantic similarity. Binary correctness is sometimes too coarse, especially for open-ended document questions. A generated answer may be partially correct, or it may express the same idea as the reference using different wording. To capture this softer notion of answer quality, we also compute semantic similarity between the generated answer and the reference answer.

Specifically, we embed both the generated answer and the reference answer, then compute their cosine similarity. Higher cosine similarity indicates stronger semantic overlap with the reference. This score is not meant to replace correctness judgment; instead, it provides a complementary view of answer quality, especially when answers are long, open-ended, or difficult to grade with a strict binary label.

We use semantic similarity mainly for Paper and Notice, where answers can be more open-ended and may vary in phrasing. For these datasets, cosine similarity helps compare whether methods recover the same meaning even when the generated text does not exactly match the reference. We analyze similarity distributions using cumulative distribution functions (CDFs), which show how often each method produces answers above different similarity thresholds.

E Additional Results

E.1 Comparison with AttentionRAG

We compare SAGE with AttentionRAG, a closely related training-free attention-based method. Unlike SAGE, AttentionRAG selects highly attended tokens and expands them to their containing sentences, so its top-k parameter does not directly control the final context size. We therefore compare the two methods at settings with nearly identical measured context usage, using SAGE at $b = 0.10$ and AttentionRAG at Top-$k = 15$.

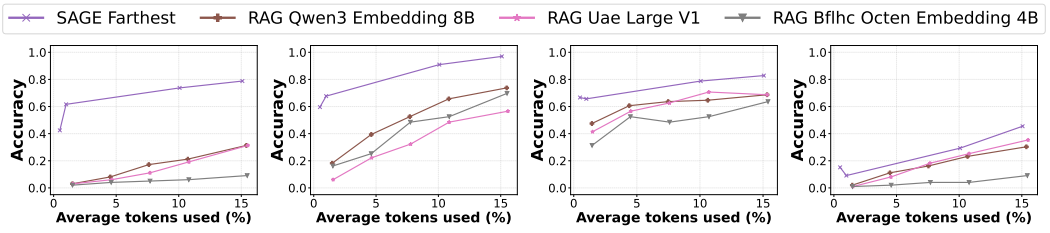
Table 10: **Comparison with AttentionRAG on QuALITY-hard.** We compare methods by measured token usage because AttentionRAG's top-k setting does not directly control final context size.

Method	Setting	Accuracy	Token Usage
AttentionRAG	Top-$k = 15$	39.62%	10.25%
SAGE (Qwen3-8B)	$b = 0.10$	**52.02%**	10.07%
SAGE (Qwen3-8B)	$b = 0.15$	57.00%	15.08%
SAGE (Qwen3-8B)	$b = 0.30$	63.10%	30.09%
SAGE (Qwen3-8B)	$b = 0.40$	64.13%	40.11%

Table 11: **Comparison with FINCH on overlapping LongBench tasks.** FINCH compresses KV-cache states, while SAGE selects original text, so their budgets are not directly comparable. SAGE uses the Fixed contrast query because LongBench lacks document-level question pools for Farthest selection.

Method	Dataset	Selection Setting	F1	Avg. Tokens
SAGE	MultiFieldQA-EN	$b = 0.15$	**54.24**	1,072.2
FINCH	MultiFieldQA-EN	2,000 tokens	44.13	2,000
SAGE	NarrativeQA	$b = 0.05$	**18.09**	1,501.5
FINCH	NarrativeQA	2,000 tokens	17.60	2,000
SAGE	QASPER	$b = 0.05$	**33.65**	253.0
FINCH	QASPER	2,000 tokens	23.19	2,000

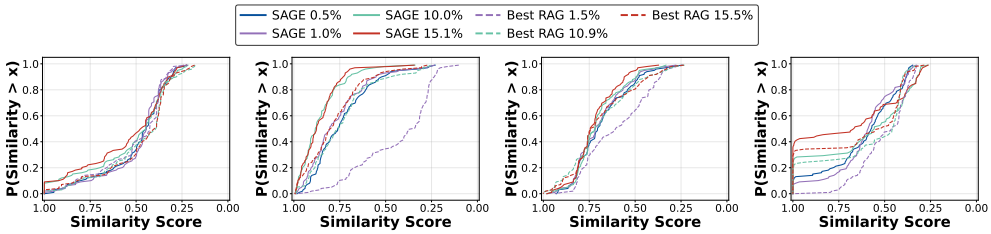
E.2 Additional Paper results.



(a) What is the number of authors (b) What is the main contribution of the paper (c) What is the publication year and venue of the paper (d) What is the title of the paper

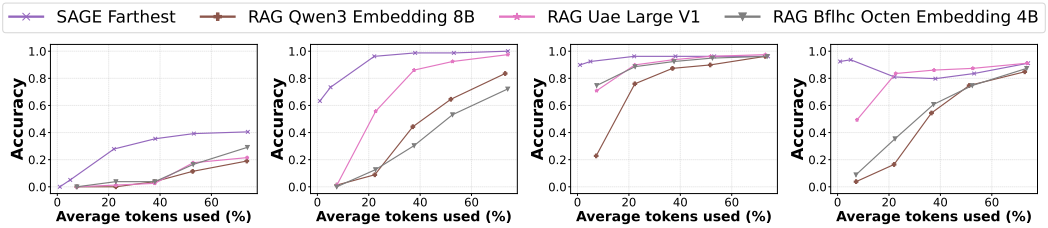
Figure 5: **LLM-judge accuracy on the Paper dataset.** Accuracy versus average token usage for SAGE and RAG baselines across representative questions.

We include representative results on the Paper dataset from ZenDB (Lin et al., 2025). Paper documents are longer and more structured than standard narrative passages, and the questions range from localized metadata queries to more semantic questions such as the main contribution of a paper.



(a) What is the number of authors (b) What is the main contribution of the paper (c) What is the publication year and venue of the paper (d) What is the title of the paper

Figure 6: **Semantic similarity CDFs on the Paper dataset.** CDFs of cosine similarity between generated and reference answers, comparing SAGE to the best RAG baseline at matched average token usage. Results use Qwen3-8B for attention estimation.



(a) Violation items (b) Notice date (c) Company name (d) State abbreviation and ZIP code

Figure 7: **LLM-judge accuracy on Notice.** Accuracy versus average context-token usage for SAGE and RAG baselines across all four questions. SAGE can operate below the size of a single RAG chunk while generally matching or outperforming RAG at comparable token usage.

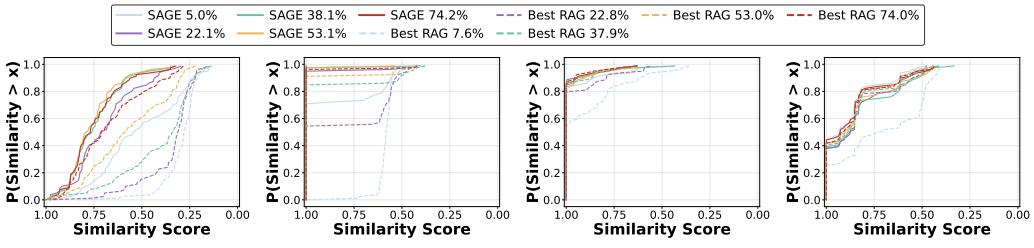
Fig. 5 shows that SAGE generally achieves stronger LLM-judge accuracy than RAG baselines at comparable token usage. Fig. 6 further shows that SAGE produces answers with higher semantic similarity to the reference across representative questions.

The Paper results also highlight the difference between metadata-style questions and more semantic questions. For localized questions such as title, author count, and publication venue, SAGE can often identify the relevant region with fewer tokens. For open-ended questions such as the main contribution, SAGE benefits from additional budget because the evidence is more distributed, but it still improves more effectively than retrieving more chunks.

E.3 Additional Notice Results

Figs. 7 and 8 show the complete Notice comparisons. Fine-grained selection supports contexts below the size of a full retrieval chunk, while generally matching or improving on RAG at comparable token usage.

A useful failure case is the question asking for the company's state abbreviation and ZIP code (Fig. 7d). Accuracy decreases as more context is included, and the similarity distributions show a corresponding pattern (Fig. 8d). One possible explanation is that larger contexts include additional address-like fields and competing ZIP-code candidates. This illustrates why retaining more context need not improve the final answer.



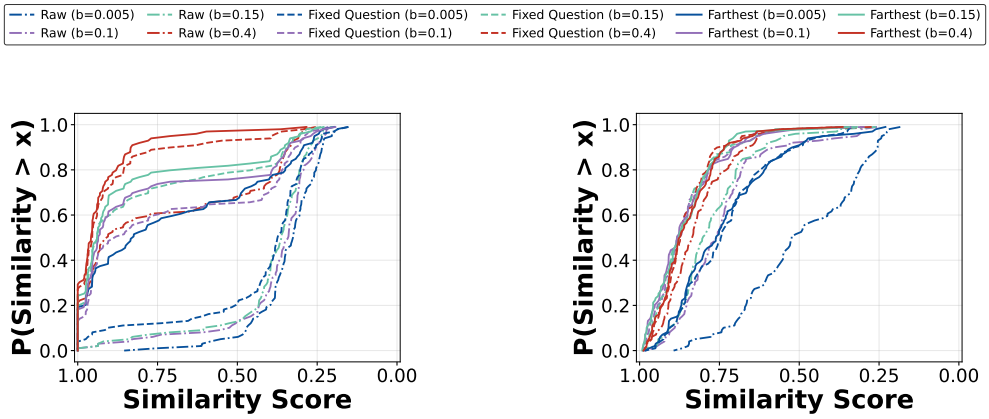
(a) Violation items (b) Notice date (c) Company name (d) State abbreviation and ZIP code

Figure 8: **Semantic-similarity distributions on Notice.** We compare SAGE with the best-performing RAG baseline at matched average context usage across all four questions. Similarity is computed between generated and reference answers using Qwen3-Embedding-0.6B. Higher curves indicate stronger reference similarity.

F Ablation Studies

F.1 Effect of Differential Attention Strategy

The end-to-end Paper comparison appears in Table 4. Fig. 9 adds a graded view through semantic-similarity distributions. Differential attention generally improves on Raw, while Farthest shows stronger semantic alignment in the reported distributions, especially for the main-contribution question. This complements, rather than overrides, the binary-accuracy result: Fixed performs best at three of the four tested Paper budgets. We use Farthest as the default contrast, not as a claim that it is best for every task, metric, or budget.



(a) Authors (b) Main contribution

Figure 9: **Effect of differential attention on Paper answer similarity.** Complementary CDFs compare Raw, Fixed, and Farthest attention using Qwen3-8B at the token budgets shown in the plots. At threshold x, the vertical axis gives the fraction of answers with reference similarity at least x, so higher curves are better. The distributions complement the end-to-end accuracy results in Table 4; better semantic alignment does not imply higher binary accuracy at every budget.

F.1.1 Effect of Contrast-Query Similarity

We additionally select the most similar available question from the same document, denoted as Closest. On QuALITY-hard, Closest improves accuracy over Raw from 51.08% to 59.53%, but remains below Fixed and Farthest (Table 12). This is consistent with similar questions removing some shared background attention while also subtracting attention to shared answer evidence. A

Table 12: **Effect of contrast-query similarity on QuALITY-hard.** Accuracy (%) with Qwen3-8B at $b = 0.30$ and GPT-4o-mini evaluation. Raw applies no contrastive subtraction. Closest and Farthest select the most and least similar available questions from the same document, respectively, while Fixed uses a fixed contrast query. Closest improves on Raw, but Farthest has the highest reported accuracy in this comparison.

Attention strategy	Accuracy (%)
Raw	51.08
Closest	59.53
Fixed	61.78
Farthest	**63.10**

Table 13: **Chunk and query normalization ablation.** Accuracy (%) on QuALITY-hard with Qwen3-8B, Farthest contrast, and $b = 0.30$, using the main generation and evaluation setup. The 2×2 design varies only the two normalization components. Keeping both gives the highest accuracy. Disabling chunk normalization, query normalization, or both reduces accuracy by 1.41, 3.29, and 4.41 percentage points, respectively.

Chunk normalization	Query normalization	
	Enabled	Disabled
Enabled	**63.10**	59.81
Disabled	61.69	58.69

more distant question is less likely to require that same evidence. When no same-document question pool is available, Fixed remains applicable.

F.2 Effect of Chunk and Query Normalization

Table 13 varies chunk and query normalization on QuALITY-hard. Removing chunk normalization reduces accuracy by 1.41 percentage points, while removing query normalization reduces it by 3.29 points. Removing both gives the largest drop, 4.41 points. With fixed-length chunks, chunk normalization mainly corrects the shorter final chunk. Query normalization aligns attention scales before subtracting maps from queries of different lengths. The results support retaining both components.

F.3 Limitations

SAGE depends on the quality of the attention signal. Our experiments show that even a small 1B selector can give useful evidence-selection signals, but stronger selectors often produce cleaner heatmaps and better selected spans. This is a limitation of the current setup, but the framework itself is flexible. The selector can be replaced by a stronger local or domain-specific model without changing the generator or retraining the method.

SAGE can also be confused by redundant or duplicate information in the document. This happens in the Notice results for the ZIP-code question. There, SAGE performs best at small budgets, but accuracy decreases as more context is included. The same trend appears in the similarity CDF in Fig. 8d. Notice documents often contain repeated address-like fields. With a small budget, SAGE usually keeps the most relevant field. With a larger budget, it may include extra ZIP-code-like candidates, which makes the final answer less precise. This shows a real limitation of attention-based selection: when several fields look similar, the method may not always know which one is the main answer. At the same time, this example also supports our main point that more context is not always better.

Finally, the current span selection step is greedy. This keeps SAGE simple and efficient, but it may not always find the best global set of evidence spans. Future work can improve this step with diversity-aware selection, adaptive window sizes, or lightweight re-ranking, while keeping the method training-free and plug-and-play.

G BROADER IMPACT

SAGE is designed to make long-context question answering more efficient by selecting a small set of query-relevant evidence before generation. This can lower inference cost and make long-document QA easier to use when passing the full context to a generator is expensive. It can also make the process more transparent, since users can inspect which parts of the document are kept.

However, any evidence-selection method may miss useful context. If the reduced context leaves out important information, the generator may produce incomplete or unsupported answers. This is especially important in high-stakes domains such as legal, medical, or financial analysis. In these settings, SAGE should be used as an assistive context-reduction tool, and the selected evidence should remain visible for verification rather than being treated as a final answer by itself.