
DSAS: A Universal Plug-and-Play Framework for Attention Optimization in Multi-Document Question Answering

Jiakai Li¹, Rongzheng Wang¹, Yizhuo Ma¹,
Shuang Liang¹, Guangchun Luo², Ke Qin^{1*}

¹Institute of Intelligent Computing,

University of Electronic Science and Technology of China, Chengdu, China

²School of Information and Software Engineering,

University of Electronic Science and Technology of China, Chengdu, China

ljk@std.uestc.edu.cn, qinke@uestc.edu.cn

Abstract

While large language models (LLMs) show considerable promise across various fields, they have notable limitations in handling multi-document question answering (Multi-doc QA) tasks. The first challenge is long-range dependency modeling, where LLMs struggle to focus on key information in long texts, which weakens important semantic connections. Second, most LLMs suffer from the “lost-in-the-middle” issue, where they have difficulty processing information in the middle of long inputs. Current solutions either truncate global dependencies or demand costly finetuning, ultimately lacking a universal and simple solution for these challenges. To resolve these limitations, we propose Dual-Stage Adaptive Sharpening (DSAS) containing two modules. (i) The Contextual Gate Weighting (CGW) module alleviates “lost-in-the-middle” by assessing paragraph relevance through layer-wise attention tracking and position-aware weighting. (ii) The Reciprocal Attention Suppression (RAS) module enhances focus on critical paragraphs by suppressing information exchange between key and irrelevant texts, thus mitigating the limitations in long-range dependency modeling. Extensive experiments on four benchmarks demonstrate DSAS’s efficacy across mainstream LLMs (Llama, Qwen, Mistral, and Deepseek), with an average F1-score improvement of 4.2% in Multi-doc QA tasks on Llama-3.1-8B-Instruct and Qwen2.5-14B-Instruct. Ablation studies confirm the essential contributions of both the CGW and RAS modules. In addition, detailed discussions in the Appendix further validate the robustness and scalability of DSAS.

1 Introduction

Transformer-based [34] large language models (LLMs) have demonstrated remarkable performance, which have extensively promoted various complex natural language processing applications [9, 25, 30, 21, 37]. Building on the progress, recent advancements have shifted research focus toward enhancing LLMs’ long-context processing capabilities [23], giving rise to LLMs that significantly expand context windows from 4K tokens to 128K or even 1M tokens, e.g., Llama-3.1-8B-Instruct [10] and Gemini-1.5 [31]. These LLMs have unlocked unprecedented potential for complex tasks requiring cross-document reasoning, such as legal case analysis [8] and multi-source academic synthesis [7].

*Corresponding author.

Code is available at <https://github.com/TreasureHunter/DSAS>

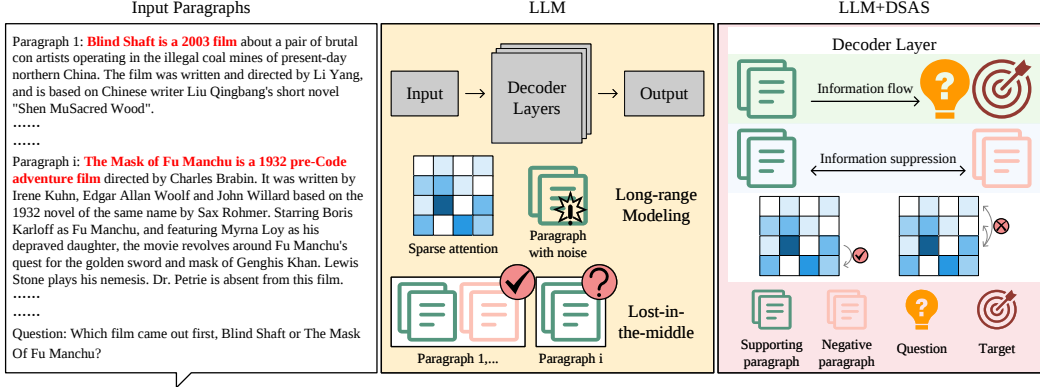


Figure 1: In Multi-doc QA tasks, directly processing long inputs comprising numerous paragraphs with LLMs presents two major challenges: long-range dependency modeling and “lost-in-the-middle”, resulting in degraded answer quality. DSAS acts as a plug-in that enhances LLMs through a dual-stage process: (i) reinforcing information flow between supporting passages and both the question and target, and (ii) suppressing interactions between supporting and negative paragraphs.

However, simply concatenating multiple documents into these long contexts often results in degraded performance due to attention dilution [27], a phenomenon where critical inter-document dependencies are overshadowed by irrelevant tokens. While solutions like StreamingLLM [40] and Selective Self-attention [44] are introduced, they either truncate global dependencies or lack generalizability, leaving the core challenge of context-aware attention prioritization unresolved.

As shown in Figure 1, the aforementioned attention dilution phenomenon reveals two critical limitations in multi-document question answering (Multi-doc QA) scenarios: (i) Limited long-range dependency modeling: Despite claims of 128K-token support, RULER [15] reveals that LLMs often struggle with real-world tasks requiring combinatorial reasoning. While recent efforts (e.g., StreamingLLM [40], LM-Infinite [13]) explore attention mechanism optimization to address the challenge, they sacrifice global token interactions, compromising Multi-doc QA performance. (ii) Persistent “lost-in-the-middle” issue: Nelson et al. [24] demonstrate that LLMs perform poorly when key information appears in the middle of long inputs. Current solutions like LongAlign [2] using a hybrid strategy (combining long-instruction examples with short data) require additional training using curated datasets, making them less adaptable to mainstream LLMs. These challenges highlight the critical need for universal, plug-and-play modules that enhance Multi-doc QA capabilities without architectural constraints or task-specific fine-tuning.

Recent studies [35, 43, 17] demonstrate that attention-driven information flow analysis provides critical insights into model reasoning patterns. Inspired by this insight, we adapt the methodology to Multi-doc QA tasks. To address these issues, we introduce Dual-Stage Adaptive Sharpening (DSAS) as shown in Figure 1, a training-free attention optimization mechanism comprising two modules: Contextual Gate Weighting (CGW) and Reciprocal Attention Suppression (RAS). Specifically, our method works as follows: CGW tracks attention scores across selected model layers and calculates a contextual gate weight for each paragraph. Additionally, CGW introduces a position-aware weighting mechanism to enhance focus on information in the middle. Then RAS identifies critical paragraphs and attenuates information exchange between critical paragraphs and irrelevant content. DSAS requires no architectural changes or extra finetuning, serving as a universal plug-in for Transformer-based LLMs to strengthen Multi-doc QA capabilities.

Our contributions are as follows: (i) We systematically investigate the information flows on several LLMs in Multi-doc QA through paragraph disparity level and answer quality level analysis. (ii) Based on the findings, we propose DSAS, a training-free universal plug-in for Transformer-based LLMs, which enhances focus on critical information through CGW and RAS, while suppressing irrelevant content. (iii) Extensive experiments on four public benchmark datasets demonstrate the effectiveness of DSAS for various LLMs including Llama, Qwen, Mistral and Deepseek, achieving an average F1-score improvement of up to 4.2% on Llama-3.1-8B-Instruct [10] and Qwen2.5-14B-Instruct [28].

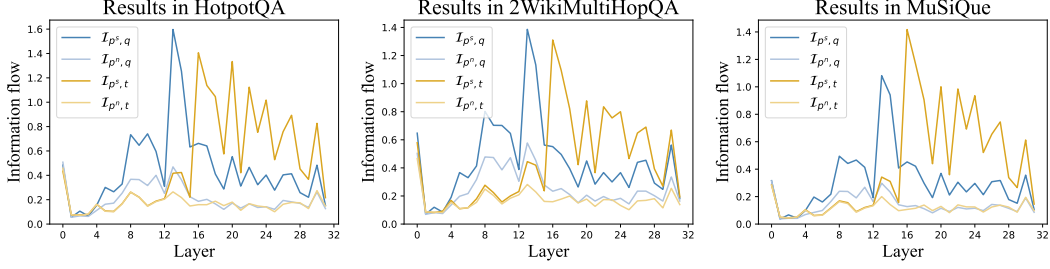


Figure 2: Layer-wise information flows of HotpotQA, 2WikiMultiHopQA and MuSiQue tested on Llama-3.1-8B-Instruct. p^s and p^n denote supporting paragraphs and negative paragraphs, respectively. The results of Qwen2.5-7B-Instruct are shown in Appendix A.2.

2 Information Flow Analysis on Multi-doc QA

Identifying key factors for effective Multi-doc QA reasoning is essential. To achieve this, we conduct a systematic analysis of LLMs’ inference processes across three core components: input paragraphs (p), question (q), and target (t). A critical step lies in selecting suitable methods to study the semantic interactions among these components. Attention score analysis [29], a method widely used to examine information flow, is adapted here to investigate how LLMs integrate cross-document information.

2.1 Preliminaries

Let $A_{h,l}^S$, $A_{h,l}^W$ denote the attention score matrix and the attention weight matrix of the h -th head in the l -th layer, respectively. We obtain the layer-specific matrices A_l^S and A_l^W by summing across all attention heads. Here, $A_l^W(i, j)$ represents the information flow from the j -th to the i -th token.

To analyze interactions among three key components: (i) paragraphs ($p^1, \dots, p^C$): Each paragraph p^m spans input token indices $\{p_s^m, p_s^m + 1, \dots, p_e^m\}$, where p_s^m and p_e^m denote the start/end positions and C is the total paragraph count; (ii) question q ; and (iii) target t : the answer generation position (i.e., the final token in the input), we propose two metrics. $\mathcal{I}_{p^m, q}$ and $\mathcal{I}_{p^m, t}$ are the Top-K significance of the information flow from the m -th paragraph p^m to q and p^m to t , respectively:

$$\mathcal{I}_{p^m, q} = \frac{1}{Q} \sum \text{Top-K} \left(\left\{ \sum_{i \in q} A_l^W(i, j) | j \in p^m \right\} \right), \quad (1)$$

$$\mathcal{I}_{p^m, t} = \sum \text{Top-K} (\{A_l^W(t, j) | j \in p^m\}), m \in \{1, \dots, C\}, \quad (2)$$

where Q denote the token length of the question. $\mathcal{I}_{p^m, q}$ and $\mathcal{I}_{p^m, t}$ serve as information flow indicators to analyze the inference process for LLMs. The larger their values are, the more the generated results pay attention to the corresponding paragraphs.

Experimental Settings. We choose Llama-3.1-8B-Instruct [10] and Qwen2.5-7B-Instruct [28] as our primary models for investigation, due to their moderate model size and strong instruction following ability. For datasets, since LongBench [3] lacks supporting facts (labels for supporting paragraphs), we use HotpotQA [41], 2WikiMultiHopQA [14] and MuSiQue [33]. We sample 1000 examples from the training set for evaluation. Templates for constructing inputs are provided in Appendix A.1.

2.2 Paragraph Disparity Level Analysis

Most LLMs adopt a multi-layer Transformer architecture, with each decoder block processing semantic information differently during inference. We intend to check layer-wise information flows of $\mathcal{I}_{p^m, q}$ and $\mathcal{I}_{p^m, t}$. Figure 2 visualizes layer-wise attention weight values. These values serve as a quantified metric to display how the different paragraphs contribute to the generated answer from the paragraph disparity perspective: Information flows from distinct paragraphs diverge only slightly in the shallow layers of the LLM, which suggests that the LLM first establishes basic semantic understanding to support deeper processing. As layers deepen, a clear divergence emerges between

$\mathcal{I}_{p^s,q}$ and $\mathcal{I}_{p^n,q}$, demonstrating the LLM progressively distinguishes task-relevant paragraphs through its layer-wise semantic processing, enabling rational attention allocation. Ultimately, in the deep layers of the LLM, LLM recognizes and utilizes key paragraphs for answer generation.

Our analysis uncovers a two-stage reasoning pattern: (i) Information initially converges on the question, with supporting paragraphs exhibiting stronger information flows than negative paragraphs, confirming the model’s ability to prioritize semantically relevant content. (ii) Subsequently, the information from supporting paragraphs aggregates to the target, where the LLM strategically leverages critical paragraphs to formulate answers. These observations highlight the inherent interpretability of LLMs, proving their capabilities to integrate task-specific information during inference.

2.3 Answer Quality Level Analysis

We define a reasoning process as GOOD if its output exactly matches the reference answer, and BAD if it contains no word-level overlap with the reference answer (see Appendix A.4 for more details). To investigate the divergence between good and bad reasoning patterns, we aggregate information flow values across all model layers and conduct a comparative analysis of two distinct groups: $\mathcal{I}_{p^s,q}$ and $\mathcal{I}_{p^s,t}$ from supporting paragraphs, along with $\mathcal{I}_{p^n,q}$ and $\mathcal{I}_{p^n,t}$ from negative paragraphs. As shown in Figure 3, good reasoning exhibits higher values of $\mathcal{I}_{p^s,q}$ and $\mathcal{I}_{p^s,t}$, and lower values of $\mathcal{I}_{p^n,q}$ and $\mathcal{I}_{p^n,t}$. The MuSiQue dataset reveals the largest gaps between good and bad reasonings across all four metrics, which is probably attributable to its multi-hop reasoning complexity.

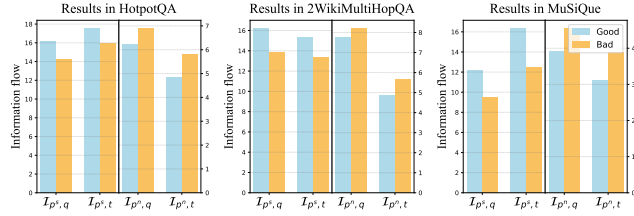


Figure 3: Comparison between mean values of the good and bad instances from the 1000 samples of Llama-3.1-8B. The results of Qwen2.5-7B are shown in Appendix A.3.

Notably, even in bad reasoning instances, $\mathcal{I}_{p^s,q}$ and $\mathcal{I}_{p^s,t}$ for supporting paragraphs consistently exceed those of $\mathcal{I}_{p^n,q}$ and $\mathcal{I}_{p^n,t}$ for negative paragraphs. This observation indicates that LLMs possess inherent discrimination capabilities between supporting and negative paragraphs regardless of reasoning quality, though the information flows of supporting paragraphs remain insufficient for optimal answer generation. Our framework addresses this core issue by introducing a quantitative information flow analysis that measures $\mathcal{I}_{p,q}$ and $\mathcal{I}_{p,t}$ for each paragraph. This analysis enables the selective and precise amplification of relevant semantic information, as detailed in Section 3.

3 Methodology

To explore the Multi-doc QA capabilities of LLMs, we analyze the attention score matrix in each layer and identify the key paragraphs through the information flows. Building on the above findings, we propose Dual-Stage Adaptive Sharpening (DSAS), which consists of two key modules. In Section 3.1, we introduce Contextual Gate Weighting (CGW), which identifies key paragraphs and strengthens the attention of the question and target position toward these texts. Section 3.2 describes Reciprocal Attention Suppression (RAS), which aims to suppress interactions between key paragraphs and irrelevant information. At a high level, we strategically integrate CGW and RAS modules into the computation of multi-head attention to implement DSAS:

$$\begin{aligned}
A_{h,l}^S &= \frac{QW_{h,l}^Q(KW_{h,l}^K)^T}{\sqrt{d_k}}, \quad A_l^S = \text{Stack}(A_{1,l}^S, \dots, A_{H,l}^S), \\
A_{h,l}^S &= \text{RAS}(\text{CGW}(A_{h,l}^S, A_l^S)), \\
A_{h,l}^W &= \text{Softmax}(A_{h,l}^S, \dim = -1), \\
O_{h,l} &= A_{h,l}^W(VW_{h,l}^V), \quad O_l = \text{Concat}(O_{1,l}, \dots, O_{H,l})W_l^O,
\end{aligned} \tag{3}$$

where $W_{h,l}^Q, W_{h,l}^K, W_{h,l}^V$ are projection matrices of the h -th head in the l -th layer, W_l^O is the output projection matrix. We STACK tensors along the first dim and CONCAT them along the last dim. The framework of DSAS is shown in Figure 4. To avoid ambiguity, descriptions of key symbols appearing in the following text are provided in Table 5 in Appendix B.

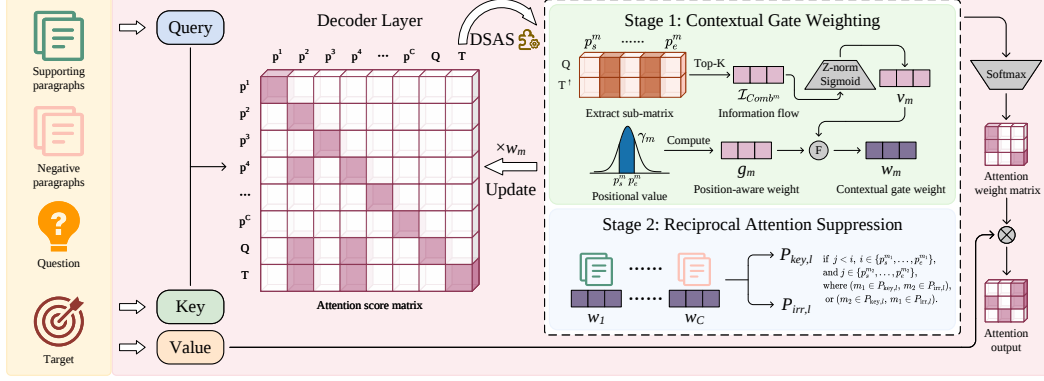


Figure 4: Illustration of Dual-Stage Adaptive Sharpening (DSAS), including Contextual Gate Weighting (CGW) and Reciprocal Attention Suppression (RAS).

3.1 Stage 1: Contextual Gate Weighting (CGW)

For each selected layer, we compute the combined information flows of each paragraph using the attention score matrix A_l^S . Specifically, we extract the attention sub-matrix corresponding to the positions of each paragraph, question and target as follows:

$$M \in \mathbb{R}^{2Q \times p^m} = \underbrace{[A_l^S(q, p^m)]}_{\text{Question Matrix}} \parallel \underbrace{[A_l^S(t, p^m) \uparrow^{(Q)}]}_{\text{Target Matrix}}, \quad (4)$$

where Q, p^m denote the token length of the question and the m -th paragraph, respectively, and $\uparrow^{(Q)}$ represents the expand operation to match the question length. The combined information flow for the m -th paragraph is then calculated by averaging the column-wise Top-K values of M :

$$\mathcal{I}_{Comb^m} = \frac{1}{K} \sum \text{Top-K} \left[\sum_{i=1}^{2Q} M_{i,j} \right]_{j=1}^{p^m}, m \in \{1, \dots, C\}. \quad (5)$$

We then compute v_m through Z-normalization and sigmoid scaling:

$$v_m = 0.5 \cdot \sigma \left(\frac{\mathcal{I}_{Comb^m} - \mu_I}{\sigma_I} \right) + 0.5, m \in \{1, \dots, C\} \quad (6)$$

where $\sigma(\cdot)$ is the sigmoid function and μ_I, σ_I are the mean and the standard deviation of $\mathcal{I}_{Comb}$. The minimum value of v_m is set to 0.5 to prevent overlooking the paragraphs excessively. In addition, to mitigate the “lost-in-the-middle” issue, we introduce position-aware weighting that assigns greater weights to central key paragraphs. Hsieh et al. [16] attribute this limitation to the inherent U-shaped attention bias in LLMs, which places more emphasis on content at the beginning and end of the sequence. We correct this bias through the probability density function (PDF) of a Gaussian distribution, which is defined as follows:

$$f(x) = \frac{1}{\sigma_p \sqrt{2\pi}} \cdot \exp \left(-\frac{(x - \mu_p)^2}{2\sigma_p^2} \right), \quad F(x) = \int_{-\infty}^x f(t) dt \quad (7)$$

where μ_p, σ_p are the mean and the standard deviation of the input token indices $\{0, 1, \dots, L-1\}$ (i.e., μ_p equals $0.5 \cdot (L-1)$, σ_p equals $\sqrt{\frac{L^2-1}{12}}$), and $F(x)$ is the cumulative distribution function (CDF) of a Gaussian distribution. Then the positional value γ_m for the m -th paragraph is computed based on the token indices $\{p_s^m, p_s^m + 1, \dots, p_e^m\}$ within the segment:

$$z_1 = \frac{p_s^m - \mu_p}{\sigma_p}, \quad z_2 = \frac{p_e^m - \mu_p}{\sigma_p}, \quad (8)$$

$$\gamma_m = \frac{F(z_2) - F(z_1)}{z_2 - z_1}, \quad (9)$$

where z_1, z_2 denote the normalized value of p_s^m and p_e^m , respectively. Next, we rank paragraphs by v_m , compute position-aware weights for the top 50% paragraphs, and assign a value of 1 to the rest, aiming to prevent the model from attracting excessive attention to non-critical middle paragraphs. The position-aware weight g_m is calculated as follows:

$$g_m = \begin{cases} \left(\frac{0.5C+1}{\text{rank}_m} \right)^{\gamma_m}, & \text{if } \text{rank}_m \leq 0.5C, \\ 1, & \text{otherwise.} \end{cases} \quad (10)$$

The final contextual gate weight w_m is derived as:

$$w'_m = v_m \cdot g_m^\alpha, \quad (11)$$

$$w_m = (1 - \beta) \frac{w'_m - \min(w'_i)}{\max(w'_i) - \min(w'_i)} + \beta, i \in \{1, \dots, C\} \quad (12)$$

where hyperparameters α and β balance positional and content relevance and determine the minimum value of w_m , respectively. The attention score matrix $A_{h,l}^S$ is dynamically adjusted by applying contextual gate weights of each paragraph, thereby refining the model's focus based on their importance:

$$A_{h,l}^S(i, j) = w_m \cdot A_{h,l}^S(i, j), \quad \text{if } i \in \{q, t\}, j \in \{p_s^m, p_s^m + 1, \dots, p_e^m\} \quad (13)$$

3.2 Stage 2: Reciprocal Attention Suppression (RAS)

The CGW module computes contextual gate weights for each paragraph to evaluate their relevance to answer generation, enabling clear distinction between critical and irrelevant paragraphs. The RAS module aims to suppress interactions between non-critical and key paragraphs. To achieve this, we classify paragraphs using a threshold-based method: paragraphs with contextual gate weights higher than the mean value of all w_m are identified as key paragraphs, while those below the threshold are labeled as irrelevant paragraphs. The key and irrelevant paragraphs are denoted as $P_{key,l}$ and $P_{irr,l}$ in the l -th layer. Next, reciprocal attention suppression is applied between these categories. This process adjusts the attention score matrix $A_{h,l}^S$ by suppressing values between key and irrelevant paragraphs. The reciprocal suppression is bidirectional, affecting both interactions between $P_{key,l}$ and $P_{irr,l}$ to break cross-paragraph interference. Formally, the suppression is implemented as:

$$\begin{aligned} A_{h,l}^S(i, j) &= \min(w_{m_1}, w_{m_2}) \cdot A_{h,l}^S(i, j), \\ \text{if } j < i, i &\in \{p_s^{m_1}, p_s^{m_1} + 1, \dots, p_e^{m_1}\}, \text{ and } j \in \{p_s^{m_2}, p_s^{m_2} + 1, \dots, p_e^{m_2}\}, \\ \text{where } (m_1 \in P_{key,l}, m_2 \in P_{irr,l}) &\text{ or } (m_2 \in P_{key,l}, m_1 \in P_{irr,l}). \end{aligned} \quad (14)$$

4 Experiments

This section assesses DSAS, a plug-and-play attention mechanism designed to enhance Multi-doc QA performance across diverse models and downstream tasks. Our experiments are structured as follows: Section 4.1 details the implementation of DSAS, covering datasets, metrics, models and hyperparameter settings. In Section 4.2, we show that DSAS achieves improvements on all Multi-doc QA benchmarks, including HotpotQA [41], 2WikiMultiHopQA [14], MuSiQue [33], and LongBench [3], without requiring any additional training. Section 4.3 presents ablation studies to examine the effectiveness of DSAS under different variants and hyperparameters. Section 4.4 further explores the robustness of DSAS.

4.1 Implementation

Datasets & Metrics. We evaluate our method on three classic Multi-doc QA benchmarks and a long-context dataset. Following previous works, we utilize the validation splits of HotpotQA [41], 2WikiMultiHopQA [14], and MuSiQue [33] to test our effectiveness, as they provide reference answers. To further assess generalization, we extend experiments to LongBench's [3] corresponding subsets on Multi-doc QA tasks. Referring to HotpotQA and LongBench, we consider the F1-score as the evaluation metric for these datasets. The metrics details are shown in Appendix A.4.

Baselines & Models. PINE [38] analyzes attention patterns to re-assign input paragraph positions. Given its similar operation and target to DSAS, we include PINE as a baseline for comparison, alongside a vanilla LLM baseline that performs inference without altering the original pipeline. We select

Table 1: Comparison results on HotpotQA, 2WikiMultiHopQA, MuSiQue and LongBench benchmarks. 2WikiMQA denotes 2WikiMultiHopQA. The evaluation for all tasks is assessed through the F1-score (%).

Models	Methods	HotpotQA	2WikiMQA	MuSiQue	LongBench			Average
					HotpotQA	2WikiMQA	MuSiQue	
Llama-3.2-3B	Vanilla	39.1	39.4	20.8	41.9	34.1	12.7	31.3
	PINE	39.0	39.6	21.3	42.4	35.3	12.5	31.6
	DSAS	40.6 (+1.5)	39.7 (+0.3)	21.2 (+0.4)	42.7 (+0.8)	35.9 (+1.8)	13.2 (+0.5)	32.2 (+0.9)
Mistral-7B	Vanilla	32.0	19.0	24.3	39.3	16.6	22.2	25.6
	PINE	32.4	19.5	24.6	40.2	16.9	24.1	26.3
	DSAS	33.8 (+1.8)	20.2 (+1.2)	26.8 (+2.5)	41.9 (+2.6)	16.6 (+0)	25.3 (+3.1)	27.4 (+1.8)
Qwen2.5-7B	Vanilla	42.3	46.0	30.5	55.1	50.3	28.1	42.1
	PINE	44.5	47.2	31.4	57.3	52.3	30.4	43.9
	DSAS	46.1 (+3.8)	49.9 (+3.9)	35.0 (+4.5)	57.7 (+2.6)	52.7 (+2.4)	33.4 (+5.3)	45.8 (+3.7)
DeepSeek-R1-8B	Vanilla	37.2	22.9	26.8	45.1	20.3	24.8	29.5
	PINE	36.8	22.6	26.5	45.2	20.1	23.9	29.2
	DSAS	39.0 (+1.8)	23.4 (+0.5)	28.0 (+1.2)	47.2 (+2.1)	21.2 (+0.8)	26.7 (+1.9)	30.9 (+1.4)
Llama-3.1-8B	Vanilla	43.6	47.3	34.6	53.3	42.6	25.4	41.1
	PINE	46.6	49.2	37.0	54.4	46.9	30.6	44.1
	DSAS	47.1 (+3.5)	50.8 (+3.5)	39.2 (+4.6)	56.5 (+3.2)	47.3 (+4.7)	32.0 (+6.6)	45.5 (+4.2)
Qwen2.5-14B	Vanilla	48.2	55.3	38.0	57.6	53.1	32.8	47.5
	PINE	50.4	57.0	41.8	59.8	56.4	36.7	50.4
	DSAS	51.8 (+3.6)	58.2 (+2.9)	43.8 (+5.8)	60.9 (+3.3)	56.1 (+3.0)	39.3 (+6.5)	51.7 (+4.2)
Qwen2.5-32B	Vanilla	48.8	60.7	42.3	58.6	48.2	35.0	48.9
	PINE	50.8	61.0	44.5	58.1	47.9	36.4	49.8
	DSAS	50.8 (+2.0)	62.2 (+1.5)	45.4 (+3.1)	59.5 (+0.9)	50.5 (+2.3)	39.9 (+4.9)	51.4 (+2.5)

six popular LLMs, including Llama-3.2-3B-Instruct [10], Mistral-7B-Instruct-v0.2 [1], Qwen2.5-7B-Instruct [28], DeepSeek-R1-Distill-Llama-8B [11], Llama-3.1-8B-Instruct [10], Qwen2.5-14B-Instruct [28], and Qwen2.5-32B-Instruct [28], since they are popular Transformer-based decoder-only LLMs, which are convenient for exploiting and analyzing inside architectures. All models are deployed with the “bfloat16” data format due to the balance between efficiency and performance. We set the generation mode to greedy-decoding for all methods with deterministic sampling parameters: do_sample=False, temperature=0, top_p=1, max_new_tokens=32 (follow the setting in LongBench [3]). This setting minimizes the impact of irrelevant confounders during inference, thereby ensuring that identical models and inputs always produce the same answers to fixed questions. All experiments are implemented using PyTorch 2.2.1, and executed on the CPU of two 32-core Intel(R) Xeon(R) @ 2.80GHz and GPU of 8× NVIDIA A800. The codebase is compatible with Python 3.12, and computations were accelerated using CUDA 12.1.

Hyperparameters. The hyperparameters in Equations 5, 11, and 12 are set to $K = 10$, $\alpha = 1$, and $\beta = 0.7$ for all benchmarks and LLMs. DSAS is applied to the final 50% layers of all LLMs.

4.2 Main Results

Table 1 presents the main experimental results, revealing three principal insights: (i) Simply replacing the original attention module with our DSAS enhances LLMs’ performance across all Multi-doc QA benchmarks without requiring additional training, achieving average F1-score gains between 0.9% and 4.2%. Llama-3.1-8B and Qwen2.5-14B achieve the most significant improvements, particularly on LongBench tasks. These results demonstrate that DSAS acts as a catalyst to enhance Multi-doc QA performance across diverse model architectures and task configurations. In contrast, although PINE [38] achieves performance gains under certain configurations, its improvements are not consistently observed. (ii) Performance improvements hold consistently across model scales ranging from 3B to 32B parameters, with medium-sized LLMs generally exhibiting greater performance gains (e.g., maximum improvement of Llama-3.1-8B and Qwen2.5-14B, followed by Qwen2.5-7B). We attribute the pattern to two main reasons. (a) Medium-sized LLMs (e.g., Llama-3.1-8B, Qwen2.5-14B) demonstrate adequate semantic understanding yet remain vulnerable to input noise, especially in long-context scenarios. DSAS overcomes this by analyzing information flows to precisely identify critical paragraphs and amplify model focus. In contrast, smaller models (e.g., Llama-3.2-3B) lack sufficient comprehension capacity, while larger models (e.g., Qwen2.5-32B) approach the task performance ceiling, leaving little room for further gains. (b) Medium-sized LLMs exhibit greater vulnerability to

Table 2: Ablation study of DSAS on HotpotQA, 2WikiMultiHopQA, MuSiQue and LongBench benchmarks. 2WikiMQA denotes 2WikiMultiHopQA. p-a denotes position-aware weight g_m in Equation (10). The evaluation for all tasks is assessed through the F1-score (%).

Models	Variants	HotpotQA	2WikiMQA	MuSiQue	LongBench			Average
					HotpotQA	2WikiMQA	MuSiQue	
Llama-3.2-3B	DSAS	40.6	39.7	21.2	42.7	35.9	13.2	32.2
	w/o CGW	40.6	39.5	20.4	43.5	34.6	12.9	31.9
	w/o RAS	39.7	39.0	21.7	42.9	34.6	11.4	31.6
	w/o p-a	39.1	36.8	19.0	42.4	34.5	10.6	30.4
Qwen2.5-7B	DSAS	46.1	49.9	35.0	57.7	52.7	33.4	45.8
	w/o CGW	45.5	48.2	33.8	56.6	50.3	31.1	44.3
	w/o RAS	45.4	47.8	32.6	56.7	51.3	32.0	44.3
	w/o p-a	44.7	50.5	33.2	56.2	49.6	30.6	44.1
Llama-3.1-8B	DSAS	47.1	50.8	39.2	56.5	47.3	32.0	45.5
	w/o CGW	46.6	48.9	38.4	55.6	46.6	30.9	44.5
	w/o RAS	45.7	49.0	38.7	54.9	46.8	31.4	44.4
	w/o p-a	45.9	49.7	37.8	56.2	46.2	30.4	44.4
Qwen2.5-14B	DSAS	51.8	58.2	43.8	60.9	56.1	39.3	51.7
	w/o CGW	49.5	57.2	40.8	58.6	55.3	38.2	50.0
	w/o RAS	50.2	57.6	41.4	59.9	54.1	38.6	50.3
	w/o p-a	51.3	58.9	43.1	61.2	55.6	39.2	51.6

the “lost-in-the-middle” phenomenon, while larger models are generally more robust to this issue. Overall, DSAS effectively unlocks the latent capabilities of diverse LLMs on Multi-doc QA tasks, regardless of their architecture or scale. (iii) The improvement varies across different benchmarks. For medium-sized LLMs (parameters 7B, 8B, and 14B), greater improvements emerge on MuSiQue (including its LongBench extensions) compared to other benchmarks. This discrepancy suggests that DSAS particularly enhances LLMs’ performance on complex tasks, where their inherent Multi-doc QA abilities can be better activated. The performance gap between HotpotQA and L-HotpotQA of these LLMs may be attributed to two factors. (a) Ansong et al. [26] identify annotation inconsistencies in HotpotQA that could result in unreliable assessments. (b) HotpotQA and L-HotpotQA only involve two-hop reasoning tasks with relatively lower complexity which are easy for these LLMs.

4.3 Ablation Studies

In this section, we conduct ablation studies to evaluate the effectiveness of each component in DSAS. Table 2 summarizes the performance of different variants on all tasks.

w/o CGW. Applying contextual gate weight w_m only in RAS to regulate paragraph interactions degrades all metrics. This indicates that enhancing information propagation from key paragraphs to both the question and the target is essential for answer generation.

w/o RAS. Results reveal that removing w_m during information aggregation lowers LLMs’ performance, suggesting that the absence of RAS allows irrelevant content in $P_{irr,l}$ to introduce noise into the semantics of key paragraphs $P_{key,l}$.

w/o p-a weight. Setting $\alpha = 0$ in Equation (11) weakens resistance to the “lost-in-the-middle” issue, confirming that adaptively weighting middle paragraphs through g_m enhances answer quality on most benchmarks. Notably, Qwen2.5-14B with this configuration achieves competitive results relative to DSAS, indicating larger LLMs inherently mitigate “lost-in-the-middle” more effectively, which aligns with the observations in LongPiBench [32]. Nevertheless, the position-aware weight generally improves performance on all benchmarks and LLMs, mitigating “lost-in-the-middle” issue.

Hyperparameter Study. The parameter K in Equation (5) controls the number of tokens that the question and target attend to within each paragraph. Lower K values risk insufficient contextual focus on the paragraphs, while higher values may degrade performance due to interference from irrelevant content. We evaluate K with 5, 10, 20 and observe that $K = 10$ achieves optimal performance of Llama-3.1-8B, as shown in Figure 5 (a). Furthermore, we investigate the optimal number of insertion layers n of DSAS. Section 2 shows minimal divergence in shallow layers. Most LLMs have layer counts in multiples of four (e.g., 28, 32). We therefore position DSAS at three types of depths: the final 25%, 50%, and 75% layers ($n = 25\%$, 50%, 75%, respectively) of the LLM. Figure 5 (b)

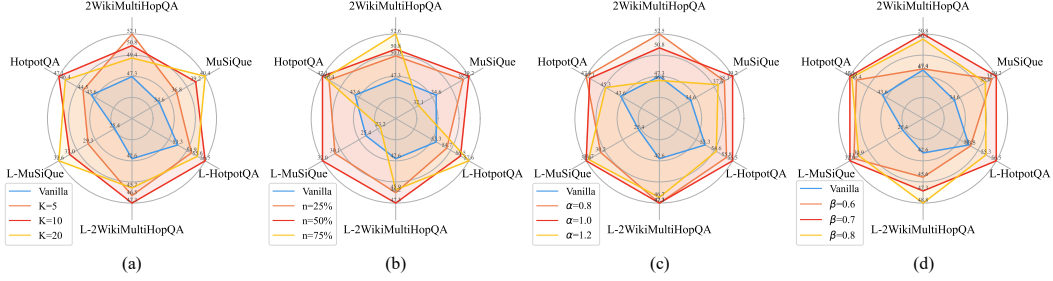


Figure 5: Llama-3.1-8B’s hyperparameter study of K , n , α , β on HotpotQA, 2WikiMultiHopQA, MuSiQue, and LongBench benchmarks. The hyperparameter studies of Llama-3.2-3B, Qwen2.5-7B, and Qwen2.5-14B are shown in Appendix C.

Table 3: Results of original and shuffled input orderings when applying DSAS.

Models	Variants	HotpotQA	2WikiMQA	MuSiQue	LongBench			Average
					HotpotQA	2WikiMQA	MuSiQue	
Qwen2.5-7B	Original	46.1	49.9	35.0	57.7	52.7	33.4	45.8
	Shuffled	46.1	49.7	35.4	57.6	53.0	33.3	45.9
Llama-3.1-8B	Original	47.1	50.8	39.2	56.5	47.3	32.0	45.5
	Shuffled	47.2	51.1	39.0	56.2	47.5	31.8	45.5
Qwen2.5-14B	Original	51.8	58.2	43.8	60.9	56.1	39.3	51.7
	Shuffled	51.7	58.3	43.8	60.9	56.0	39.3	51.7

demonstrates that best performance is achieved with DSAS at the final 50% layers position. This could be because $n = 25\%$ inadequately improves information flows, while $n = 75\%$ causes misjudgment of critical content due to the slight divergence among information flows of different paragraphs in LLMs’ shallow layers. Finally, we discuss the most appropriate values for α in Equation (11) and β in Equation (12), which are used to balance v_m and g_m and determine the minimum value of w_m , respectively. Figure 5 (c)(d) demonstrate $\alpha = 1$ and $\beta = 0.7$ are the best choices.

4.4 Further Analysis on Different Subsets

Since DSAS enhances attention on centrally located critical paragraphs, thereby mitigating the LLMs’ inherent “lost-in-the-middle” issue. It is essential to evaluate its robustness to different input orders. To this end, we conduct an experiment in which we randomize the ordering of paragraphs in all test samples while deliberately increasing the probability of supporting paragraphs occurring at the edges. This shuffled configuration intentionally weakens the attention of DSAS to these important information. As shown in Table 3, results under both settings (using the fixed configuration $K = 10$, $n=50\%$, $\alpha = 1$, and $\beta = 0.7$) reveal only minor performance fluctuations across ordering conditions. This stability demonstrates the robustness of DSAS to input order, primarily benefitting from our weighting strategy. Specifically, when evidence appears at the edges (where g_m in Equation (10) is reduced), DSAS attenuates the adjustments to the model’s native attention distribution. This design intentionally leverages the model’s inherent capacity to identify and amplify information flows from edge-positioned evidence, which aligns with the “U-shaped attention bias” phenomenon [16].

In addition, we further analyze how paragraph counts affect DSAS. Since the MuSiQue dataset contains 2-4 supporting paragraphs per sample, we group results by supporting paragraph count and report the comparative results in Table 4. We observe minimal performance gaps across the groups, indicating that DSAS better captures relevant information flows to improve accuracy. To examine the effect of total paragraph count, we select all samples from MuSiQue and construct inputs with 10, 20, 30, and 40 paragraphs. The 20-paragraph setting uses the original sample inputs; the 10-paragraph setting randomly removes ten of the negative paragraphs; the 30-paragraph and 40-paragraph settings add 10/20 paragraphs sampled from other examples. Table 4 shows that performance generally declines as the total paragraph count increases. However, DSAS degrades more slowly, since it suppresses information flow from negative paragraphs.

Table 4: Results on different subsets of MuSiQue. #Sup, #T denote supporting paragraph count and total paragraph count, respectively.

Models	Variants	#Sup=2	#Sup=3	#Sup=4	#T=10	#T=20	#T=30	#T=40
Qwen2.5-7B	Vanilla	30.4	30.9	30.3	32.3	30.5	29.8	29.4
	DSAS	35.0	35.2	34.7	36.7	35.0	34.8	34.3
Llama-3.1-8B	Vanilla	35.8	32.9	33.6	35.4	34.6	33.8	32.0
	DSAS	40.2	35.7	36.5	39.5	39.2	38.7	38.3
Qwen2.5-14B	Vanilla	38.1	37.3	38.7	39.3	38.0	36.9	36.1
	DSAS	44.0	43.3	44.3	44.7	43.8	43.5	42.9

5 Background and Related Works

Long-context Reasoning in LLMs. Recent advances in LLMs have spurred the development of long-context models (e.g., Llama-3.1-8B-Instruct [10] and Qwen2.5-7B-Instruct [28], claiming 128k-token capacities). However, mainstream capacity benchmarks like Needle-in-a-Haystack (NIAH) [19] mainly assess simple retrieval tasks, insufficiently reflecting complex real-world applications such as Multi-doc QA requiring cross-document information aggregation and multi-step reasoning. Current methods to enhance LLMs’ long-context reasoning abilities follow two directions: (i) Positional encoding extensions (e.g., PI [6], CLEX [4], and CREAM [39]) effectively expand context windows by adjusting position embeddings, but demand extra training resources and struggle to integrate with existing LLMs with inherently large context windows. (ii) Attention mechanism optimizations reduce computational costs and support longer inputs with minimal additional training. However, most approaches prioritize efficiency over actual reasoning improvements. Both strategies leave significant gaps in enhancing LLMs’ reasoning abilities within their designated context windows.

Attention mechanism. The Transformer [34] architecture LLMs currently face two challenges: quadratic computational and memory scaling with input length and the “lost-in-the-middle” [24] phenomenon. To overcome these constraints, recent studies are proposed to optimize the standard attention framework, including StreamingLLM [40], LM-Infinite [13], H2O [45], PINE [38], and SSC [42]. StreamingLLM and LM-Infinite optimize memory usage by retaining initial tokens and recent tokens for next-token prediction, balancing efficiency and performance. H2O preserves recent tokens while dynamically selecting critical tokens (termed heavy hitters) to balance local relevance and global importance. PINE employs bidirectional inter-segment attention and re-assigns paragraph positions. SSC scales hidden states to enable more balanced attention distribution across different segments. Collectively, current approaches either truncate long contexts to reduce computational and memory demands (e.g., StreamingLLM, LM-Infinite, H2O) or overlook the long-range modeling among paragraphs (e.g., PINE, SSC). Therefore, we propose DSAS, a training-free framework that adaptively optimizes attention matrices. Other approaches (e.g., prompt compression [18], retrieval-augmented generation [20], memory tree [5]) require external models or higher computation.

6 Conclusion

In this work, we address the critical challenges of limited long-range modeling and persistent “lost-in-the-middle” issue in Multi-doc QA for LLMs. Through a systematic analysis of information flow patterns from the paragraph disparity level and the answer quality level, our study reveals layer-wise aggregation patterns in distinct information flows and differences between good and bad reasoning instances. To resolve these issues, we propose Dual-Stage Adaptive Sharpening (DSAS), a training-free, plug-and-play mechanism that adaptively sharpens attention focus through two synergistic components: Contextual Gate Weighting (CGW) and Reciprocal Attention Suppression (RAS).

This work highlights the unexploited potential of attention optimization for LLMs. By adaptively sharpening attention score matrices, DSAS offers a practical solution for real-world applications requiring cross-document reasoning. Future directions include extending DSAS to diverse long-context scenarios, exploring its integration with retrieval-augmented generation frameworks (e.g., RAG [20], GraphRAG [9]), and investigating related LLM applications (i.e., backdoor attacks [12], data distillation [22], graph reasoning [36]). Our findings underscore the importance of optimal attention optimization in unlocking the full capabilities of modern LLMs for complex tasks.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (Grant No.62406057 and No.62176046), the Fundamental Research Funds for the Central Universities (Grant No.ZYGX2025XJ042), the Noncommunicable Chronic Diseases-National Science and Technology Major Project (Grant No.2023ZD0501806), and the Sichuan Science and Technology Program (Grant No.2024ZDZX0011). We gratefully acknowledge the Advanced Computing Frontier Research Center of Tianfu Jiangxi Laboratory for providing the computing resources that enabled this research, which has been successfully deployed in the laboratory’s nurse agent.

References

- [1] Mistral AI. Mistral-7b-instruct-v0.2, 2024. URL <https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2>. Huggingface.
- [2] Yushi Bai, Xin Lv, Jiajie Zhang, Yuze He, Ji Qi, Lei Hou, Jie Tang, Yuxiao Dong, and Juanzi Li. LongAlign: A recipe for long context alignment of large language models. In *EMNLP*, pages 1376–1395. Association for Computational Linguistics, 2024.
- [3] Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. Longbench: A bilingual, multitask benchmark for long context understanding. In *ACL*, pages 3119–3137, Bangkok, Thailand, 2024. Association for Computational Linguistics.
- [4] Guanzheng Chen, Xin Li, Zaiqiao Meng, Shangsong Liang, and Lidong Bing. CLEX: continuous length extrapolation for large language models. In *ICLR*, Vienna, Austria, 2024. OpenReview.net.
- [5] Howard Chen, Ramakanth Pasunuru, Jason Weston, and Asli Celikyilmaz. Walking down the memory maze: Beyond context limit through interactive reading. *arXiv preprint arXiv:2310.05029*, 2023.
- [6] Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. Extending context window of large language models via positional interpolation. *arXiv preprint arXiv:2306.15595*, 2023.
- [7] Maxime Darrin, Ines Arous, Pablo Piantanida, and Jackie Chi Kit Cheung. GLIMPSE: pragmatically informative multi-document summarization for scholarly reviews. In *ACL*, pages 12737–12752, Bangkok, Thailand, 2024. Association for Computational Linguistics.
- [8] Chenlong Deng, Kelong Mao, and Zhicheng Dou. Learning interpretable legal case retrieval via knowledge-guided case reformulation. In *EMNLP*, pages 1253–1265, Miami, FL, USA, 2024. Association for Computational Linguistics.
- [9] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitan, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*, 2024.
- [10] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.
- [11] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.
- [12] Ji Guo, Wenbo Jiang, Rui Zhang, Wenshu Fan, Jiachen Li, Guoming Lu, and Hongwei Li. Backdoor attacks against hybrid classical-quantum neural networks. *Neural Networks*, 191: 107776, 2025.
- [13] Chi Han, Qifan Wang, Hao Peng, Wenhan Xiong, Yu Chen, Heng Ji, and Sinong Wang. Lm-infinite: Zero-shot extreme length generalization for large language models. In *NAACL*, pages 3991–4008, Mexico City, Mexico, 2024. Association for Computational Linguistics.

- [14] Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing A multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *COLING*, pages 6609–6625, Barcelona, Spain, 2020.
- [15] Cheng-Ping Hsieh, Simeng Sun, Samuel Krizan, Shantanu Acharya, Dima Rekish, Fei Jia, Yang Zhang, and Boris Ginsburg. Ruler: What’s the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*, 2024.
- [16] Cheng-Yu Hsieh, Yung-Sung Chuang, Chun-Liang Li, Zifeng Wang, Long T. Le, Abhishek Kumar, James R. Glass, Alexander Ratner, Chen-Yu Lee, Ranjay Krishna, and Tomas Pfister. Found in the middle: Calibrating positional attention bias improves long context utilization. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 14982–14995. Association for Computational Linguistics, 2024.
- [17] Qidong Huang, Xiaoyi Dong, Pan Zhang, Bin Wang, Conghui He, Jiaqi Wang, Dahua Lin, Weiming Zhang, and Nenghai Yu. OPERA: alleviating hallucination in multi-modal large language models via over-trust penalty and retrospection-allocation. In *CVPR*, pages 13418–13427, Seattle, WA, USA, 2024. IEEE.
- [18] 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 *ACL*, pages 1658–1677, Bangkok, Thailand, 2024. Association for Computational Linguistics.
- [19] Greg Kamradt. Needle in a haystack test, 2023. URL https://github.com/gkamradt/LLMTest_NeedleInAHaystack. GitHub repository.
- [20] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *NeurIPS, December 6-12, 2020, virtual*, 2020.
- [21] Muquan Li, Dongyang Zhang, Tao He, Xiurui Xie, Yuan-Fang Li, and Ke Qin. Towards effective data-free knowledge distillation via diverse diffusion augmentation. In *MM, Melbourne, VIC, Australia, 28 October 2024 - 1 November 2024*, pages 4416–4425. ACM, 2024.
- [22] Muquan Li, Hang Gou, Dongyang Zhang, Shuang Liang, Xiurui Xie, Deqiang Ouyang, and Ke Qin. Beyond random: Automatic inner-loop optimization in dataset distillation. *arXiv preprint arXiv:2510.04838*, 2025.
- [23] 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. *arXiv preprint arXiv:2503.17407*, 2025.
- [24] 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. *Trans. Assoc. Comput. Linguistics*, 12:157–173, 2024.
- [25] Yizhuo Ma, Ke Qin, and Shuang Liang. Beta-LR: Interpretable logical reasoning based on beta distribution. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1945–1955, Mexico City, Mexico, June 2024. Association for Computational Linguistics.
- [26] Ansong Ni, Matt Gardner, and Pradeep Dasigi. Mitigating false-negative contexts in multi-document question answering with retrieval marginalization. In *EMNLP*, pages 6149–6161, Virtual Event / Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics.
- [27] Zhen Qin, Xiaodong Han, Weixuan Sun, Dongxu Li, Lingpeng Kong, Nick Barnes, and Yiran Zhong. The devil in linear transformer. In *EMNLP*, pages 7025–7041, Abu Dhabi, United Arab Emirates, 2022. Association for Computational Linguistics.

- [28] Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL <https://arxiv.org/abs/2412.15115>.
- [29] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *ICLR*, Banff, AB, Canada, 2014.
- [30] Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel M. Ni, Heung-Yeung Shum, and Jian Guo. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *ICLR*, Vienna, Austria, 2024. OpenReview.net.
- [31] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.
- [32] Runchu Tian, Yanghao Li, Yuepeng Fu, Siyang Deng, Qinyu Luo, Cheng Qian, Shuo Wang, Xin Cong, Zhong Zhang, Yesai Wu, et al. Distance between relevant information pieces causes bias in long-context llms. *arXiv preprint arXiv:2410.14641*, 2024.
- [33] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Trans. Assoc. Comput. Linguistics*, 10:539–554, 2022.
- [34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, Long Beach, CA, USA, 2017.
- [35] Lean Wang, Lei Li, Damai Dai, Deli Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. Label words are anchors: An information flow perspective for understanding in-context learning. In *EMNLP*, pages 9840–9855, Singapore, 2023. Association for Computational Linguistics.
- [36] Rongzheng Wang, Shuang Liang, Qizhi Chen, Yihong Huang, Muquan Li, Yizhuo Ma, Dongyang Zhang, Ke Qin, and Man-Fai Leung. Graphcogent: Mitigating llms’ working memory constraints via multi-agent collaboration in complex graph understanding, 2025. URL <https://arxiv.org/abs/2508.12379>.
- [37] Rongzheng Wang, Shuang Liang, Qizhi Chen, Jiasheng Zhang, and Ke Qin. Graphtool-instruction: Revolutionizing graph reasoning in llms through decomposed subtask instruction. In *KDD, Toronto, ON, Canada, August 3-7, 2025*, pages 1492–1503. ACM, 2025.
- [38] Ziqi Wang, Hanlin Zhang, Xiner Li, Kuan-Hao Huang, Chi Han, Shuiwang Ji, Sham M. Kakade, Hao Peng, and Heng Ji. Eliminating position bias of language models: A mechanistic approach. In *ICLR, Singapore, April 24-28, 2025*. OpenReview.net, 2025.
- [39] Tong Wu, Yanpeng Zhao, and Zilong Zheng. An efficient recipe for long context extension via middle-focused positional encoding. In *NeurIPS*, Vancouver, BC, Canada, 2024.
- [40] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *ICLR*, Vienna, Austria, 2024. OpenReview.net.
- [41] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *EMNLP*, pages 2369–2380, Brussels, Belgium, 2018. Association for Computational Linguistics.
- [42] Yijiong Yu, Huiqiang Jiang, Xufang Luo, Qianhui Wu, Chin-Yew Lin, Dongsheng Li, Yuqing Yang, Yongfeng Huang, and Lili Qiu. Mitigate position bias in llms via scaling a single hidden states channel. In *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria*, pages 6092–6111. Association for Computational Linguistics, 2025.

- [43] Xiaosong Yuan, Chen Shen, Shaotian Yan, Xiaofeng Zhang, Liang Xie, Wenxiao Wang, Renchu Guan, Ying Wang, and Jieping Ye. Instance-adaptive zero-shot chain-of-thought prompting. In *NeurIPS*, Vancouver, BC, Canada, 2024.
- [44] Xuechen Zhang, Xiangyu Chang, Mingchen Li, Amit K. Roy-Chowdhury, Jiasi Chen, and Samet Oymak. Selective attention: Enhancing transformer through principled context control. In *NeurIPS*, Vancouver, BC, Canada, 2024.
- [45] Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, and et al. H2O: heavy-hitter oracle for efficient generative inference of large language models. In *NeurIPS*, New Orleans, LA, USA, 2023.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope?

Answer: [\[Yes\]](#)

Justification: The abstract and introduction clearly outline the core contributions of the paper: (1) systematic analysis of information flow in LLMs for Multi-doc QA tasks. (2) proposal of Dual-Stage Adaptive Sharpening (DSAS), a training-free plug-and-play framework with two modules: Contextual Gate Weighting (CGW) for alleviating “lost-in-the-middle” and Reciprocal Attention Suppression (RAS) for long-range dependency modeling optimization. (3) extensive experiments demonstrating DSAS’s universality across diverse LLMs (average 4.2% F1-score improvement on Llama-3.1-8B-Instruct and Qwen2.5-14B-Instruct). These align with the analysis (Section 2), methodology (Section 3) and experimental results (Section 4).

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: Appendix G explicitly discuss limitations, including DSAS’s limited applicability to other long-context tasks and its high memory and computational costs when handling extremely lengthy inputs.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.

- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[Yes\]](#)

Justification: Our framework's theoretical foundation about LLMs' information flow patterns is detailed in Section 2. Figures 2, 3 provide quantitative evidence of LLMs' information flow patterns.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [\[Yes\]](#)

Justification: The paper provides comprehensive experimental details necessary for reproducibility. Section 2 describes the experimental configuration for information flow analysis on the Multi-doc QA tasks, including dataset specifications and model implementations. Supplementary details such as prompt templates are provided in Appendix A.1. In Section 4.1, datasets (HotpotQA, 2WikiMultiHopQA, MuSiQue, LongBench), model architectures (Llama, Qwen, Mistral, Deepseek variants), and hyperparameter settings (K , n , α , β) are explicitly defined. Furthermore, the implementations of "bfloat16" deployment and greedy-decoding are used to ensure consistency.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Datasets (HotpotQA, 2WikiMultiHopQA, MuSiQue and LongBench) are publicly available with citations provided in Section 2 and Section 4.1. License information for all datasets is included in references.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Section 2 and Section 4.1 describe datasets, models, hyperparameter settings (K , n , α , β), and experimental environments. Appendix A.1 and Appendix A.4 illustrate experimental templates and assessment method, respectively.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [\[Yes\]](#)

Justification: The information flow analysis in Section 2 are conducted with 1000 test instances per task. To ensure the universality of our DSAS, we incorporate multiple LLMs (Llama, Qwen, etc.) in our experiments in Section 4. Appendix D provides additional analysis of error cases.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [\[Yes\]](#)

Justification: The hardware specifications, including GPU type and memory, are detailed in Section 4.1. Additionally, the computational complexity of our method is discussed in Appendix E.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: The work focuses on attention optimization for Multi-doc QA tasks without ethical risks. All datasets are publicly available for academic use, adhering to their original licenses.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [No]

Justification: The paper focuses on methodological contributions for attention optimization mechanism in the Multi-doc QA tasks. Broader impacts (e.g., misuse potential) are minimal as the work is foundational.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: This work uses publicly available LLMs (Llama, Qwen, etc.) and benchmark datasets (HotpotQA, 2WikiMultiHopQA, etc.). The paper does not introduce new pretrained language models, image generators, or scraped datasets that pose inherent misuse risks. All experiments involve standard academic datasets and LLMs cited in Section 4.1, posing no special safety risks requiring safeguards.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [\[Yes\]](#)

Justification: All datasets (HotpotQA, 2WikiMultiHopQA, etc.) and LLMs (Llama, Qwen, etc.) are cited with original references (Section 4.1). Publicly available datasets and LLMs are used under their respective licenses.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [\[NA\]](#)

Justification: No new datasets, models, or codebases are released. The paper focuses on attention optimization methodology for enhanced Multi-QA performance on existing benchmarks (HotpotQA, 2WikiMultiHopQA, etc.). Experimental implementations will be shared post-acceptance following anonymity guidelines.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: All experiments use publicly available Multi-doc QA datasets. No human subjects, crowdsourced data, or participant compensation are involved in this research.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: This work does not involve human subjects, biological data, or sensitive personal information. The research exclusively analyzes anonymized text data from academic benchmarks, exempt from IRB oversight per institutional guidelines.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: The paper proposes DSAS, which operates as a plug-and-play attention optimization mechanism applied to existing Transformer-based LLMs for enhanced Multi-doc QA performance. The LLMs are the basis of the methodology.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

A Details of Information Flow Analysis on Multi-doc QA

A.1 Experimental Templates

Consistent with LongBench [3], the templates for HotpotQA [41], 2WikiMultiHopQA [14], and MuSiQue [33] remain the same. Each data instance includes a question and its context, with task instructions positioned at both the start and end of the prompt to enhance model comprehension. This template is maintained consistently throughout our experiments in Section 4.

Answer the question based on the given paragraphs. Only give me the answer and do not output any other words.
The following are given paragraphs.
{context}
Answer the question based on the given paragraphs. Only give me the answer and do not output any other words.
Question: {question}
Answer:

A.2 Paragraph Disparity Level Analysis on Qwen

Qwen2.5-7B consists of 28 stacked decoder layers. Analytical results on Figure 6 demonstrate similar conclusions to Llama-3.1-8B, with minimal information flow variations in shallow layers, followed by emerging divergences between $\mathcal{I}_{p^s,q}$ and $\mathcal{I}_{p^n,q}$, and finally gaps between $\mathcal{I}_{p^n,q}$ and $\mathcal{I}_{p^n,t}$. Notably, Qwen2.5-7B exhibits a slower progression of these divergences compared to Llama-3.1-8B, with the HotpotQA and the MuSiQue datasets particularly showing no significant information flow variations across the first 50% of the model’s layers.

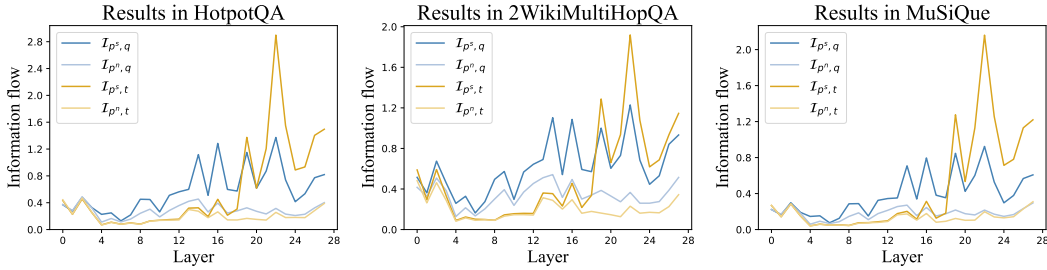


Figure 6: Layer-wise information flows of HotpotQA, 2WikiMultiHopQA and MuSiQue tested on Qwen2.5-7B. p^s and p^n denote supporting paragraphs and negative paragraphs, respectively.

A.3 Answer Quality Level Analysis on Qwen

Similar to observations in Llama-3.1-8B, significant differences in information flow patterns emerge between good and bad reasoning instances, a consistent phenomenon observed in LLMs that forms the basis of DSAS framework.

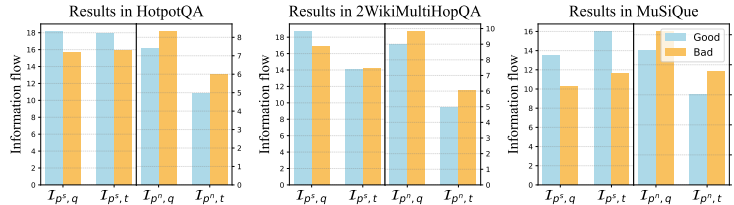


Figure 7: Comparison between mean values of the good and bad instances from the 1000 samples of Qwen2.5-7B.

A.4 Answer Quality Assessment

The evaluation method employed in these Multi-doc QA benchmarks adopts the answer quality assessment approach from HotpotQA. Specifically, for each data instance, both the reference answer and response first undergo normalization, including lowercase conversion of all textual content, elimination of non-essential articles (i.e., a/an/the), and redundant whitespace. Then precision and F1-score are computed between the normalized reference answer and response through exact lexical matching. This assessment ensures comparability across different benchmarks. We classify samples achieving a perfect F1-score (equal to 1) as GOOD reasoning cases, and assign those with zero precision (equal to 0) to the BAD reasoning instances. The assessment method remains the same for experiments at Section 4.

B Symbol Explanation

Table 5: Symbols and Descriptions.

Symbol	Description
$A_{h,l}^S$	the attention score matrix of the h -th head in the l -th layer
$A_{h,l}^W$	the attention weight matrix of the h -th head in the l -th layer
$\mathcal{I}_{comb}^m$	the combined information flow for the m -th paragraph
μ_I, σ_I	the mean and the standard deviation of $\mathcal{I}_{comb}$
v_m	the normalized value of $\mathcal{I}_{comb}^m$
μ_p, σ_p	the mean and the standard deviation of sequence $\{0, 1, \dots, L-1\}$
γ_m	the positional value of the m -th paragraph
g_m	the position-aware weight of the m -th paragraph
$w_m, \bar{w}_m$	the temporary and the final contextual gate weight of the m -th paragraph
K	the number of tokens selected to compute $\mathcal{I}_{comb}^m$
n	the proportion of layers inserting DSAS
α, β	hyperparameters for computing w_m

C Full Hyperparameter Study

The results presented in Figures 8, 9, 10 indicate that DSAS maintains consistent improvement across diverse LLMs under the hyperparameter configuration in Section 4.1 (red regions in the radar charts). Furthermore, we observe that the robustness of DSAS to hyperparameter variations strengthens with LLMs’ scale expansion (3B, 7B, 14B), with Qwen2.5-14B particularly showing notable and stable performance improvements across all benchmarks under different hyperparameter settings. We believe that this phenomenon correlates with LLMs’ scales and their intrinsic capabilities. Additionally, some hyperparameter configurations of DSAS may marginally degrade performance below baseline for Llama-3.2-3B, which further validates that our method achieves maximum performance gains specifically for middle-sized LLMs (7B, 8B, 14B).

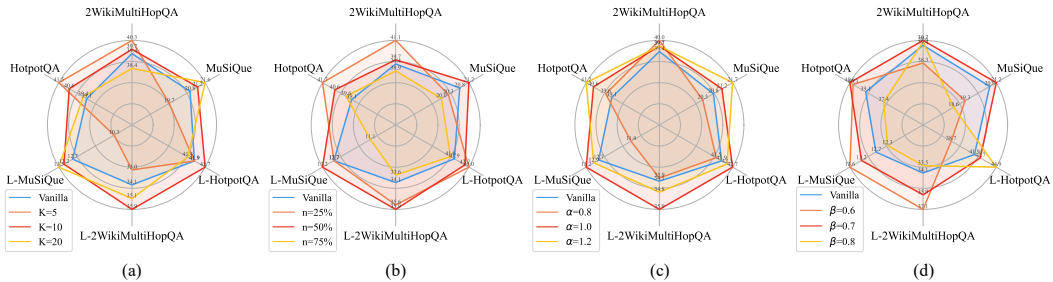


Figure 8: Hyperparameter study of K , n , α , β on HotpotQA, 2WikiMultiHopQA, MuSiQue, and LongBench benchmarks of Llama-3.2-3B.

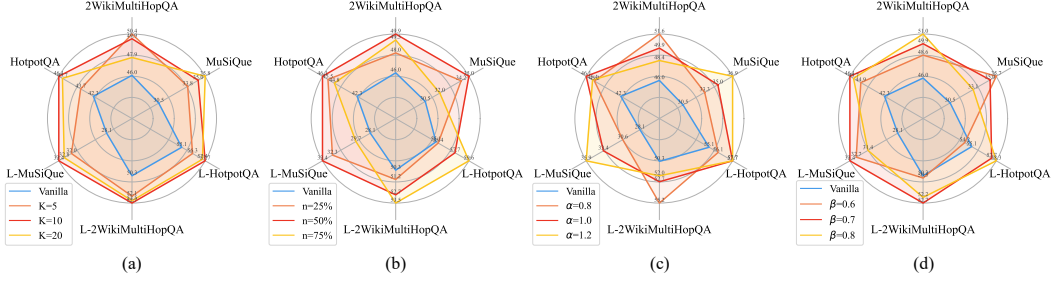


Figure 9: Hyperparameter study of K , n , α , β on HotpotQA, 2WikiMultiHopQA, MuSiQue, and LongBench benchmarks of Qwen2.5-7B.

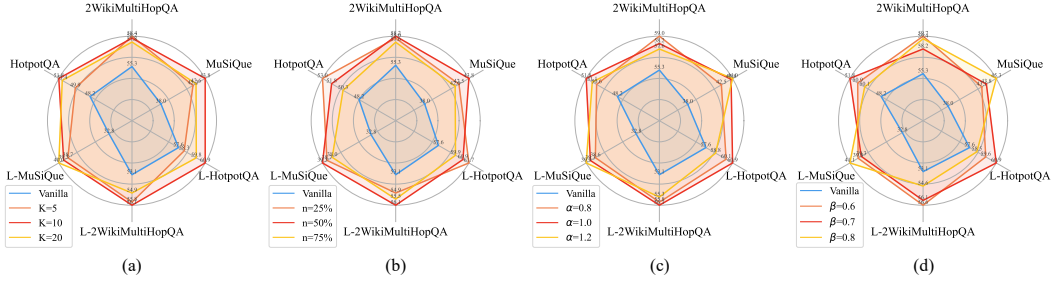


Figure 10: Hyperparameter study of K , n , α , β on HotpotQA, 2WikiMultiHopQA, MuSiQue, and LongBench benchmarks of Qwen2.5-14B.

D Error Analysis

We conduct an error analysis of DSAS with a focus on attention weight interactions among paragraphs (supporting and negative paragraphs), questions, and target. Following the experimental setup in Section 2, we randomly selected 1,000 samples for confusion matrix visualization between components. The analysis is conducted on the HotpotQA dataset due to its standardized structure of fixed 10-paragraph inputs (including 2 supporting paragraphs), which facilitates stable observation of interaction patterns.

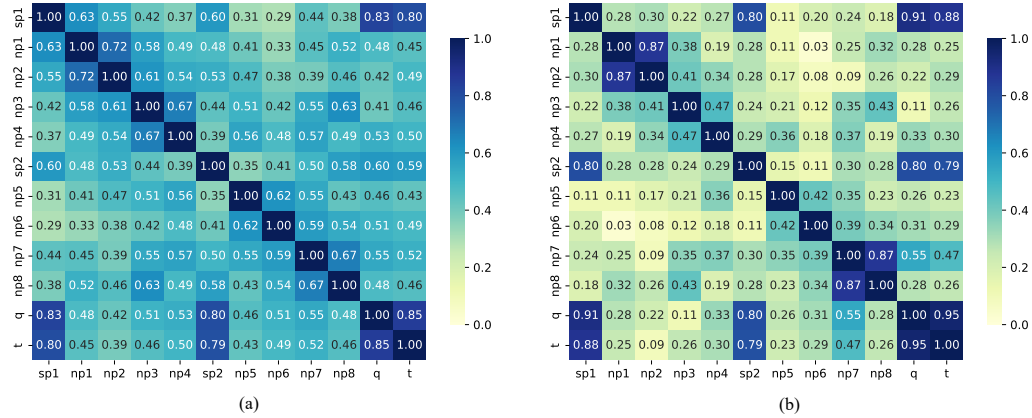


Figure 11: Confusion matrices on HotpotQA. (a) LLM, (b) LLM+DSAS. sp1, sp2 represent two supporting paragraphs. np1, np2, ..., np8 represent eight negative paragraphs. q and t denote question and target, respectively. The results are conducted on Llama-3.1-8B-Instruct.

Technically, we first average attention weights across all layers to obtain a global attention matrix. For pairwise component analysis (e.g., between two paragraphs), we extract corresponding sub-matrix based on token index ranges. All component pairs are analyzed following this method. Similar to Equation 5, we average the column-wise Top-K values for each sub-matrix. Finally we normalize all confusion values to 0-1. This approach effectively captures salient attention patterns while minimizing noise interference. Figure 11 presents heatmap comparisons of reasoning processes using LLM and LLM+DSAS. The results demonstrate that LLM+DSAS not only strengthens the focus of questions and targets on relevant support paragraphs, but also enhances information interactions between these supporting paragraphs. Additionally, it effectively suppresses unnecessary interactions between supporting paragraphs and irrelevant negative paragraphs.

E Complexity Analysis

Table 6 presents the complexity of DSAS. Here, L denotes the input token count, Q denotes the question token count, and C is the number of paragraphs. As illustrated in Equation 3, After computing the query, key, value matrices (Q , K , V) and attention scores $A_{h,l}^S$ through traditional attention mechanisms, we refine $A_{h,l}^S$ using the CGW and RAS modules. The updated scores are normalized via softmax to derive attention weights, which are then used to generate the final attention output. The core innovation of DSAS lies in computing contextual gate weight for dynamically weighting each paragraph. The time complexity scales linearly with L and Q , contributing only a minor fraction of the total computation compared to the quadratic $\mathcal{O}(L^2)$ complexity of standard attention mechanisms. Furthermore, the space complexity grows linearly with C alone. These characteristics ensure that DSAS maintains high efficiency in practice.

Table 6: Complexity analysis of DSAS.

Time Complexity	$\mathcal{O}(LQ)$
Space Complexity	$\mathcal{O}(C)$

F Scalability

Evaluating the scalability of DSAS beyond multi-document QA presents an interesting research direction. To this end, we apply DSAS to several long-context tasks from LongBench [3], including summarization (GovReport, QMSum, MultiNews) and code completion (LCC, RepoBench-P). The success of our information flow analysis largely relies on identifying anchors, which aggregate critical information from redundant contexts to guide answer generation. While in Multi-doc QA, the question and target naturally serve as anchors, extending DSAS to broader long-context tasks requires defining appropriate anchors for each scenario. After reviewing the prompt templates, we select the final instruction sentence or the query and the final sentence as anchors for aggregating information flows (e.g., “Now, write a one-page summary of the report.\n Summary:” for GovReport; “Next line of code:” for LCC).

Table 7: Comparison results on summarization and code completion tasks with hyperparameter configuration “ $K=10$, $n=50\%$, $\alpha=1$, $\beta=0.7$ ”. The evaluation metrics for L-GovReport, L-QMSum, L-MultiNews and L-LCC, L-RBP are Rouge-L score and Edit Sim, respectively. RBP denotes RepoBench-P.

Models	Variants	L-GovReport	L-QMSum	L-MultiNews	L-LCC	L-RBP	Average
Qwen2.5-7B	Vanilla	33.6	22.4	23.7	53.7	48.2	36.3
	DSAS	37.0	25.1	26.2	56.3	51.5	39.2
Llama-3.1-8B	Vanilla	34.9	24.8	27.1	58.1	50.8	39.1
	DSAS	37.4	26.6	29.3	60.0	52.5	41.2
Qwen2.5-14B	Vanilla	38.4	28.2	31.8	63.9	55.6	43.6
	DSAS	41.0	30.4	34.6	66.7	58.9	46.3

Since DSAS operates on paragraph-level segments, which Multi-doc QA inherently provides, we employ fixed-length token segmentation for the other tasks. Specifically, we use 500-token segments for GovReport, QMSum, and RepoBench-P, and 200-token segments for MultiNews and LCC, ensuring a medium number (5-40) of paragraphs per example. The comparison results are presented in Table 7. Based on our experimental results, we make two key observations: (i) DSAS consistently enhances performance across all five tasks under three medium-sized model architectures; and (ii) these improvements generalize to diverse tasks (summarization and code completion), demonstrating that DSAS is effective beyond multi-doc QA.

The prompt templates used for these tasks align with those in the LongBench [3] benchmark. The sentences in red in the following template boxes serve as anchors.

The template for L-GovReport is shown below:

You are given a report by a government agency. Write a one-page summary of the report.
Report:
{context}
Now, write a one-page summary of the report.
Summary:

The template for L-QMSum is shown below:

You are given a meeting transcript and a query containing a question or instruction. Answer the query in one or more sentences.
Transcript:
{context}
Now, answer the query based on the above meeting transcript in one or more sentences.
Query: {input}
Answer:

The template for L-MultiNews is shown below:

You are given several news passages. Write a one-page summary of all news.
News:
{context}
Now, write a one-page summary of all the news.
Summary:

The template for L-LCC is shown below:

Please complete the code given below.
{context}
Next line of code:

The template for L-RepoBench-P is shown below:

Please complete the code given below.
{context}{input}
Next line of code:

G Limitations

While DSAS acts as a plug-and-play attention mechanism and consistently improves performance in Multi-doc QA tasks, our work has two primary limitations that warrant discussion. (1) The scalability of DSAS warrants further investigation. While our experiments on summarization and code completion tasks demonstrate consistent performance gains, the current fixed-token-count chunking strategy remains relatively simple. Future work could explore more refined and semantically-aware chunking methods for these tasks, which may further enhance information flow between key semantic segments and the generated answer. (2) Although DSAS enhances performance within standard LLM context windows, it inherits the fundamental computational and memory limitations of conventional Transformer [34] architectures, with quadratic complexity in both memory usage and

computational cost. Consequently, our approach does not address the scalability challenges associated with processing extremely long documents (e.g., those exceeding 100K tokens), where memory and computational demands become prohibitive. Future investigations could consider integrating sparse attention strategies or memory-efficient architectures to mitigate these constraints.

These limitations highlight important directions for future research while not diminishing the effectiveness of DSAS. The existing framework establishes a reliable base for advancing attention optimization in long-context processing.