

Pay Attention to What You Need

Yifei Gao* Shaohong Chen* Lei Wang† Ruiting Dai Ziyun Zhang Kerui Ren Jiaji Wu Jun Cheng

Abstract

Although large language models (LLMs) have achieved significant success in natural language processing, they still struggle with long-context comprehension. Traditional approaches to mitigating this issue typically rely on fine-tuning or retraining, which is both resource-intensive and challenging to deploy in lightweight industrial settings. In this paper, we investigate the potential to accomplish this without any additional resources. Through an in-depth study of the attention mechanism in LLMs, we propose a method called **Scaled ReAttention (SRA)** to strengthen LLMs’ ability to interpret and retrieve information by strategically manipulating their attention scores during inference. Through extensive experiments, we demonstrate that integrating SRA significantly boosts LLMs’ performance on a variety of downstream tasks, highlighting its practical potential for enhancing language understanding without incurring the overhead of traditional training.

1. Introduction

Large language models (LLMs) with attention mechanisms (OpenAI, 2023) have achieved tremendous success across a wide range of downstream tasks in recent years. Their success can largely be attributed to the superiority of the attention architecture (Vaswani et al., 2017). However, as tasks become more complex and the required contextual understanding increases, LLMs often fall short.

When the input length exceeds a certain limit, LLMs often “forget” previously mentioned content or experience “memory confusion,” leading to incorrect outputs. Even with prompt engineering techniques like Chain of Thought (CoT) (Nye et al., 2022; Wei et al., 2022), the models still struggle with complex problems. This limitation originates inherently from the model itself, making it unavoidable through fine-tuning or retraining—both of which demand substantial resources. This inspired the motivation for this paper: *enhancing the model’s comprehension and retrieval capabilities without additional training*.

We began by identifying the attention mechanism as the

critical component for retrieving and interpreting context within LLMs. Building on our empirical findings and existing research (Wang et al., 2020; Zandieh et al., 2023), we noted that most tokens—and their corresponding attention scores—have a negligible effect on the model’s reasoning. Even after eliminating the majority of these scores, the model’s performance remained nearly unchanged, as illustrated in Figure 1. Intuitively, if we can better utilize the “wasted” attention scores, the model should achieve improved performance. By manually adjusting attention scores during inference and accepting a slight trade-off in model stability, we achieved a significant improvement in comprehension and retrieval capabilities, all without any fine-tuning, retraining, or auxiliary resources. To the best of our knowledge, **this represents the first effort to address these challenges from such a perspective.**

In this paper, we introduce **Scaled ReAttention (SRA)**, a technique that first discards unimportant attention scores and then redirects them toward more informative tokens. During this process, SRA strategically relaxes the model’s inherent stability, leveraging the elimination results to further enhance its comprehension. Our technique is plug-and-play and could be integrated into a wide range of existing LLMs. With SRA, we successfully improved the performance of LongChat-7B-16K and LLaMA-3-8B on the LongChat retrieval task by over 10% compared to the original models. Additionally, we significantly outperformed the original models with LLaMA-3-8B-Instruct and LLaMA-2-13B-Chat on the XSUM summarization task. Furthermore, on the public datasets such as LongBench v1 (v2), we improved the performance of a series of LLMs by above 1.5%.

Our contribution can be concluded as follows:

- A comprehensive analysis of the attention mechanism and attention scores in LLMs, offering foundational insights into the SRA technique.
- A novel plug-and-play method that enhances the comprehension and retrieval capabilities of LLMs without the need for fine-tuning or retraining.
- Empirical evidence from extensive experiments showcasing SRA’s ability to significantly improve performance in a variety of tasks.

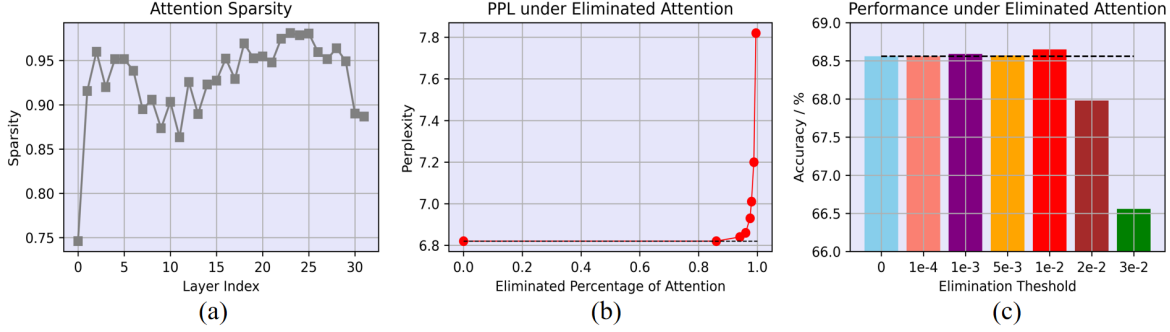


Figure 1. Characteristics of attention in LLaMA-3-8B: (a) The sparsity level of attention in each layer, with the sparsity threshold set at 0.001 on text length 2048. (b) The perplexity of WikiText2 on text length 2048 after attention elimination. (c) Averaged performance on downstream tasks (ARC, PIQA, Hellaswag, Winogrande) after attention elimination. Even with 25% of attention weights eliminated (threshold $2e-2$), the performance remains nearly unchanged. The black dashed line represents the original performance.

2. Related Work

2.1. Strengthen Long-Context Comprehension

Prior work has primarily shown how better training methods (Zhang et al., 2021; Wang et al., 2023) or larger datasets (Hoffmann et al., 2024; OpenAI, 2024) can be used to improve model performance. Despite promising results, their excessive reliance on human and computational resources imposes significant limitations on their industrial applications.

On the other hand, solving relevant issues by retrieval (Izacard et al., 2023; Jiang et al., 2022) to locate the main content while discarding irrelevant information can be equally effective. However, these approaches often require additional training of a “retriever” (Karpukhin et al., 2020) to assist with retrieval and are powerless when addressing problems that demand improved model understanding.

2.2. Extend Context Window

Previous research has highlighted the critical role of positional encoding (PE) in model performance (Vaswani et al., 2017; Su et al., 2024; Ni et al., 2022), as PE conveys essential information about the relationships between tokens. However, this adaptability can introduce substantial disruption when handling text that exceeds the model’s pre-training length (Press et al., 2022). To address this, methods such as Position Interpolation (PI) (Chen et al., 2023; Emozilla, 2023) have been proposed to extend RoPE by creating intermediate angles. Meanwhile, LandMark Attention (Mohtashami & Jaggi, 2024) incorporates an additional “Landmark” token for block-wise information representation, which slightly modifies the underlying model structure.

Although these approaches effectively broaden the context window of LLMs without introducing extensive additional resources, their achievements are at the expense

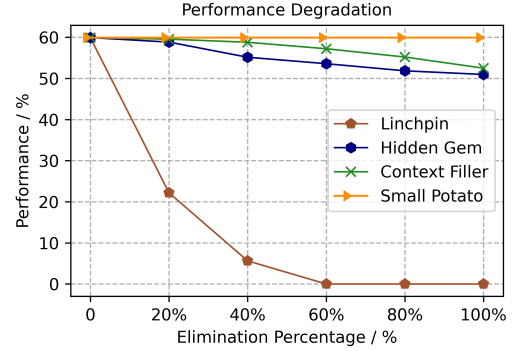


Figure 2. Performance degradation of LLaMA-2-7B-Chat after attention elimination on five LongBench tasks. Tokens with attention scores exceeding 0.05 are classified as Linchpins, those with scores in the 0.01–0.05 range (depending on their position) as Context Fillers or Hidden Gems, and those below 0.01 as Small Potatoes.

of the model’s performance on downstream tasks, which severely limits their practical applications. However, with the method proposed in this paper, their performance can be substantially improved.

3. Preliminaries

Attention Mechanism Given the input token embeddings as $\mathbf{X} \in \mathbb{R}^{n \times d}$, the attention mechanism in transformers can be computed as:

$$\text{Softmax} \left(\mathbf{Q}\mathbf{K}^\top / \sqrt{C} \right) \mathbf{V} = \mathbf{DAV} \quad (1)$$

where $\mathbf{Q} = \mathbf{X}\mathbf{W}_q$, $\mathbf{K} = \mathbf{X}\mathbf{W}_k$, $\mathbf{V} = \mathbf{X}\mathbf{W}_v$ are Query, Key, Value matrices, C is a scaling factor, and $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v \in \mathbb{R}^{d \times d}$ are projection matrices. Since Softmax can be regarded as a dynamic nonlinear scaling of KV similarity \mathbf{A} , we can use $\mathbf{D} \in \mathbb{R}^{d \times d}$ to integrate C and Softmax for a direct representation, where \mathbf{D} is dependent

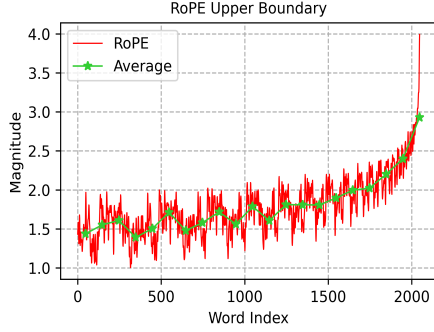


Figure 3. RoPE upper boundary alongside its averaged counterpart at intervals of 100-word index.

on \mathbf{A} .

Rotary Position Embedding Transformer models require explicit positional information to be injected. We only consider RoPE (Su et al., 2024) here, which is frequently used in many LLMs (Touvron et al., 2023; Jiang et al., 2023). Given a position index $m \in [0, c)$ and $\mathbf{X} := [x_0, x_1, \dots, x_d]^\top$, RoPE defines a vector-valued complex function $\mathbf{f}(\mathbf{X}, m)$ as follows:

$$\mathbf{f}(\mathbf{X}, m) = \begin{bmatrix} (x_0 + ix_1)e^{im\theta_0}, \\ \dots, (x_{d-2} + ix_{d-1})e^{im\theta_{d/2-1}} \end{bmatrix}^\top \quad (2)$$

where $i := \sqrt{-1}$ is the imaginary unit and $\theta_j = 10000^{-2j/d}$. In conjunction with Eq. 1, we can also integrate RoPE into a changing coefficient matrix $\mathbf{P} \in \mathbb{R}^{d \times d}$ to achieve scaling determined by relative positions:

$$\text{Softmax}(\text{Re}\langle \mathbf{f}(\mathbf{Q}, m), \mathbf{f}(\mathbf{V}, n) \rangle) = \text{DPA} \quad (3)$$

After this change, \mathbf{D} is dependent on both \mathbf{P} and \mathbf{A} .

4. Methodology

In this chapter, we first present our reasoning process and then introduce our method. We provide intuitive and easy-to-understand reasoning in the main text, with more analyses available in the appendix.

4.1. Analysis

Tokens Play Different Roles Through our experiments and analyses, we *first* defined that tokens in attention mechanisms can be categorized into 4 types, and their effects on performance after elimination are shown in Figure 2.

Linchpins \mathbf{X}_{lin} : Tokens with significantly high attention scores. These tokens often appear near the current token or the first token (Xiao et al., 2024) and frequently account for over 70% of the accumulated attention scores. These tokens often have a critical impact on the model’s reasoning results, as they are the primary contributors to altering hidden

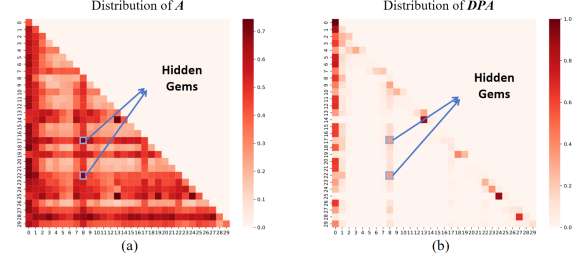


Figure 4. (a) Normalized distribution of \mathbf{A} . (b) Normalized distribution of DPA . We aim to identify those Hidden Gems (blue boxes) with high similarity at distant positions.

states between layers under the residual structure (Liu et al., 2024a) and causing outliers (Bondarenko et al., 2023).

Context Fillers \mathbf{X}_{con} : Tokens near the current token and exhibit relatively high attention scores. They generally account for approximately 25% of the total accumulated attention scores but with constrained maximum value. Their presence has only a limited impact on generation results, as the model’s reasoning capability is affected (not large) only when a large amount of them are eliminated.

Hidden Gems \mathbf{X}_{hid} : Tokens located in distant regions yet exhibiting noticeably higher attention scores. Despite their distance from the current token, these tokens exert a more pronounced impact on performance than \mathbf{X}_{con} .

Small Potatoes \mathbf{X}_{pot} : The vast majority of tokens with sparse attention. Contribute generally nothing, with accumulated attention scores no more than 2%.

Intuitively, identifying \mathbf{X}_{hid} to enhance the model’s retrieval ability is a reasonable approach. These hidden gems are expected to have high relevance with the current token, but their influence is significantly constrained due to the effects of RoPE and Softmax. Specifically, the sublinear decay ratio of RoPE at greater distances (Figure 3) combined with the exponential scaling of Softmax results in \mathbf{X}_{hid} , despite their high similarity, only barely maintaining the magnitude of attention scores as \mathbf{X}_{con} after the scaling of DP , as shown in Figure 4. From a mathematical perspective, leveraging the properties of RoPE and softmax, the classification of these four types of tokens corresponds to four distinct scaling behaviors of attention scores \mathbf{A} in both positional and magnitude spaces, as elaborated in Appendix ??.

Information Is Transferred Step by Step Under the combined effects of RoPE and softmax, attention cannot focus on tokens that are very distant from the current token, making it impossible to directly access information from distant tokens. We conducted experiments to measure how accumulated attention scores on keywords change across layers with

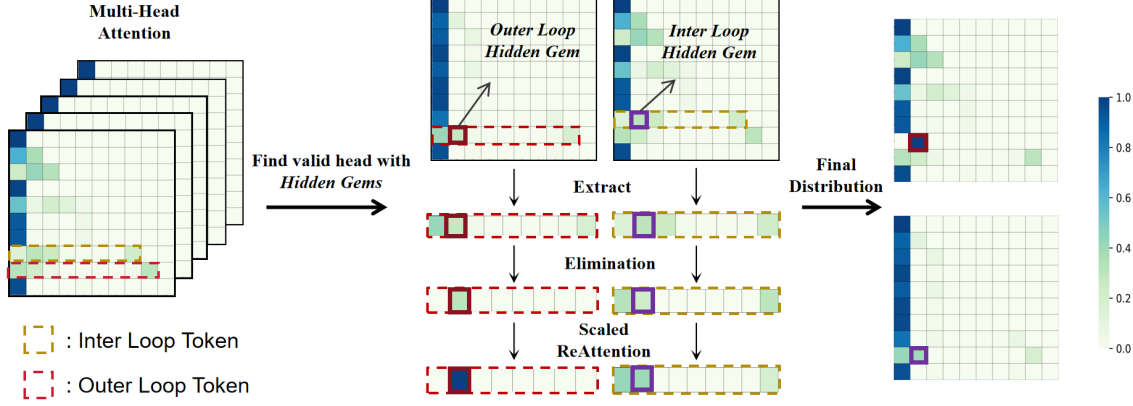


Figure 5. **Overall Pipeline.** SRA first identifies heads where the inter/outer loop contains Hidden Gems and then extracts them for attention elimination. The eliminated attention scores will be amplified (Scaled) and redistributed to these Hidden Gems (ReAttention).

varying distances. We found that beyond a certain distance, the accumulated attention score on keywords becomes minimal, yet the model is still able to produce normal outputs. A reasonable explanation for this is that information is continuously propagated during the inter-layer propagation process, ultimately being received by the current token. See more details in the Appendix ??

LLMs Are Inherently Stable In line with our attention elimination approach and previous findings, we observed that removing 30% of the accumulated attention scores across all tokens in all layers of LLMs—including most \mathbf{X}_{con} and \mathbf{X}_{hid} , as well as all \mathbf{X}_{pot} —still allowed the model to output content stably, albeit with some performance degradation on complex tasks, as shown in Figure 2. Therefore, a *moderate increase in attention scores* should also not lead to large disturbances. Based on our previous analysis, by manually identifying and amplifying \mathbf{X}_{hid} , we obtained exciting results: the model’s information comprehension and retrieval abilities improved significantly for long texts! This insight was pivotal in driving the creation of SRA.

4.2. Scaled ReAttention

Based on all the analyses above, we designed the **Scaled ReAttention (SRA)** technique with two loops: an inter-loop and an outer-loop. The inter-loop is responsible for reinforcing the transfer of information, achieved by selecting an intermediate subset of tokens and strengthening their connection with preceding tokens. Meanwhile, the outer-loop helps the final subset of tokens ignore the distance constraints introduced by PE, allowing them to allocate attention to distant tokens directly. Both loops first identify the regions to enhance. Then, they eliminate the majority of the attention weights among the selected tokens. The erased attention weights are amplified and redistributed to those

Hidden Gems within the region. The overall framework is illustrated in Figure 5.

Note that the fundamental difference between our technique and previous ones lies in the fact that, after the softmax, we increase the attention sum of certain tokens—originally limited to 1—to **exceed 1 through SRA**. These additional, intentionally introduced attentions help improve the model’s performance.

Identify Strengthened Blocks Specifically, given an attention weight matrix $\mathbf{W}_A = \mathbf{DPA} \in \mathbb{R}^{n \times n}$, we first divide it into blocks and apply the SRA operations only to specific blocks and regions. Due to the impact of the attention sink (Xiao et al., 2024), we preserve the integrity of the first C_s initial tokens. For the last C_e tokens, we specify that they only participate in the outer loop. To strategically enhance distant hidden gems, for the remaining intermediate tokens, we divide them evenly into $l + 3$ distinct blocks, where $\mathbf{C}_m^{l+3} = [C_m^1, \dots, C_m^{l+3}]$ and C_m^i is the initial token’s index for the i th block in \mathbf{C}_m^{l+3} .

The inter-loop SRA begins at layer 0 and ends at the penultimate layer, while the outer-loop SRA starts from the second layer and also ends at the penultimate layer. For every layer, both of them select only one block each. Given the selection algorithms Pick_{in} and Pick_{ou} for inter-loop and outer-loop respectively, the regions of attention weights selected for the i th layer (starting at 0th) are as follows:

$$\begin{aligned} \mathbf{W}_A[\text{Pick}_{in}(\mathbf{W}_A, i)] &= \mathbf{W}_A^{C_m^{i+4}:C_m^{i+5}, C_m^{i+1}:C_m^{i+3}} \\ \mathbf{W}_A[\text{Pick}_{ou}(\mathbf{W}_A, i)] &= \mathbf{W}_A^{-C_e:, C_m^{i+3}:C_m^{i+4}} \end{aligned} \quad (4)$$

The indexing rules here are consistent with the indexing rules of `torch.tensor` in **PyTorch**. This hierarchical approach ensures that the inter loop focuses on refining intermediate regions, while the outer loop further consolidates

and enhances these refined regions in the subsequent layer.

Attention Elimination and Scaled Redistribution The goal of elimination is to remove the smaller Context Fillers and Small Potatoes among the enhanced tokens while preserving the Hidden Gems as much as possible. Specifically, for the j th enhanced block $\mathbf{W}_{in} = \mathbf{W}_A^{C_m^j:C_m^{j+1}}$ for inter loop, $\mathbf{W}_{ou} = \mathbf{W}_A^{-C_e::}$ for outer loop, the inter-loop eliminator E_{in} and outer-loop eliminator E_{ou} is defined as:

$$\begin{aligned} E_{in}(\mathbf{W}_{in}, j) &= \text{Whe}(\mathbf{W}_{in} > (\tau_{in}/C_m^j), \mathbf{W}_{in}, 0) \\ E_{ou}(\mathbf{W}_{ou}) &= \text{Whe}(\mathbf{W}_{ou} > (\tau_{ou}/C_r), \mathbf{W}_{ou}, 0) \end{aligned} \quad (5)$$

Here, Whe functions the same as *torch.where* and $C_r = n - C_e$. If no Hidden Gems are found during the elimination process, such as all $\mathbf{W}_{in}^{C_m^{j-3}:C_m^{j-2}} = 0$, the elimination will be skipped, and no subsequent operations will be performed. Otherwise, the erased weights will be summed in a token-wise manner and multiplied by a scaling factor, s_{in} for inter loop and s_{ou} for the outer loop. This amplification enhances performance by sacrificing the stability of the LLM, allowing the accumulated attention to exceed 1. Finally, the amplified erased weights will be evenly re-added on those uneliminated Hidden Gems within targeted blocks in Eq. 4. The inter-loop algorithm is exhibited in Algorithm 1, while the outer-loop one is in the Appendix ?? . During inference, SRA is triggered *only in the prefilling stage*.

Algorithm 1 Inter-loop Scaled ReAttention

After applying Softmax on attention weights:

Input: Attention Weights \mathbf{W}_A , Layer Index i , Layer Num l , Inter Threshold τ_{in} , Inter Scaling Factor s_{in} . **{Note:}** All unspecified functions are from **PyTorch**.}

if $(l - 1) > i > 0$ **then** {Inter Loop}

/* Function only on indexes having Hidden Gems */

$\text{idx}_{gem} = \text{any}(\mathbf{W}_A[\text{Pick}_{in}(\mathbf{W}_A, i)] > (\tau_{in}/C_m^{i+4}))$

$\mathbf{W}_{eli} = E_{in}(\mathbf{W}_A[\text{idx}_{gem}], i + 4)$

$\text{idx}_{tar} = \text{Pick}_{in}(\mathbf{W}_{eli}, i)$

$\mathbf{W}_{tar} = \mathbf{W}_{eli}[\text{idx}_{tar}]$

/* Prepare scaled attention removal */

$\mathbf{W}_{re} = \text{sum}(\mathbf{W}_{eli}, \text{dim} = -1)$

$\mathbf{W}_{rm} = \text{oneslike}(\mathbf{W}_{re}) - \mathbf{W}_{re}$

$\mathbf{m}_{gem} = \text{where}(\mathbf{W}_{tar} > 0, 1, 0.01)$

$\mathbf{W}_{add} = \text{div}(\mathbf{W}_{rm}, \text{sum}(\mathbf{m}_{gem}, \text{dim} = -1)) * s_{in}$

/* Readded to original weights */

$\mathbf{W}_{eli}[\text{idx}_{tar}] = \mathbf{W}_{tar} + \mathbf{W}_{add}$

$\mathbf{W}_A[\text{idx}_{gem}] = \mathbf{W}_{eli}$

end if

5. Experiments

5.1. Setting

Our experiments comprehensively demonstrate the effectiveness of our method from multiple perspectives. We selected commonly used model series such as LLaMA (Touvron et al., 2023), Mistral (Jiang et al., 2023), Qwen (Bai et al., 2023a), and LongChat (Li et al., 2023), as well as their YaRN (Peng et al., 2023) and LandMark (Mohtashami & Jaggi, 2024) variants, as baseline models. First, we used methods from LandMark (Mohtashami & Jaggi, 2024) and LongChat (Li et al., 2023) to evaluate the improvement in retrieval capabilities brought by SRA. Next, we tested the model’s ability to summarize and understand long, complex texts on the XSUM (Narayan et al., 2018) dataset under GPT-4 evaluation protocol (Chiang et al., 2023). We further validated the superiority of our approach through downstream tasks on publicly available long-text comprehension benchmarks, including LongBench (Bai et al., 2023b), LongBench v2 (Bai et al., 2024), InfiniteBench (Zhang et al., 2024).

The configuration of SRA is not a one-size-fits-all solution. Instead, it requires dynamic tuning based on the requirements of specific tasks. Several factors influence the choice of SRA parameters, including task characteristics and variations among different baselines. In our experiments, we typically keep the total accumulated attention of SRA-strengthened tokens within the range of 1.1 to 1.4. Detailed discussions can be found in the Appendix ?? .

5.2. Reterieval Evaluation

We began by assessing the improvements in retrieval capabilities introduced by SRA within the LongChat framework, followed by an evaluation using a retrieval prompt proposed in LandMark. We modified the original retrieval prompt to increase complexity. For the *PASS KEY*, we randomly generated 50 words comprising numbers and uncommon vocabulary. By varying the retrieval distance, we tested the model’s performance. Throughout the experiments, a was fixed at 8, while b was varied at intervals of 200 tokens. The prompt and results of the two tasks are shown in Figure 6.

Experiments reveal a significant enhancement in the model’s retrieval capabilities after incorporating SRA. For the LandMark *PASS KEY* retrieval task, **LLaMA-2-7B** achieves an average improvement of 4.7% over the original model. Additionally, compared to the LandMark variant of **LLaMA-7B**, our approach delivers an average improvement of 8.5%, effectively enabling SRA to mitigate the decline in retrieval performance caused by its structure-altering. On the LongChat benchmark, SRA achieves a notable performance boost, with an average retrieval accuracy improvement exceeding 10% over these original models.

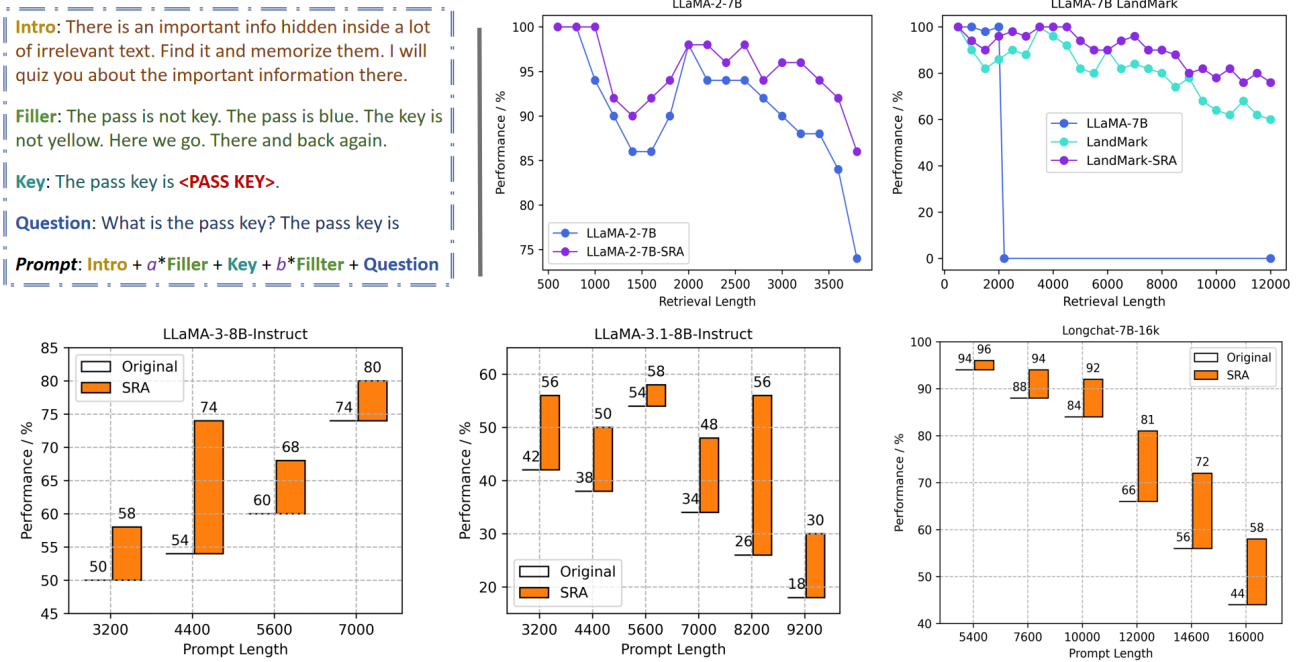


Figure 6. Retrieval Results. (Top Left) Modified Landmark retrieval prompt. Here, a and b are scaling factors used to adjust the retrieval distance. (Top Right) The retrieval results on LandMark. (Bottom) The retrieval results on LongChat. Our results indicate that SRA substantially improves retrieval capabilities across various models and tasks, highlighting its versatility and effectiveness.

Table 1. **LongBench Results.** We present the results of three open-source models evaluated on seven tasks from LongBench, both before and after applying SRA. SRA delivers consistent performance improvements without requiring any additional fine-tuning or retraining.

Model / Tasks \uparrow	MFQA-EN	VCSUM	TREC	SAMSum	LSHT	LCC	RepoBench-P	Average
LLaMA-2-7B-Chat	36.22	15	64.5	40.7	17.75	58.50	52.45	40.73
SRA	37.83	21.0	66.5	42.31	19.00	59.36	53.23	42.74(+2.01)
LLaMA-3-8B-Instruct	41.50	14.8	75.5	42.48	24.25	58.87	50.73	44.01
SRA	42.71	17.5	78.5	43.04	28.00	59.72	51.27	45.82(+1.81)
LongChat-v1.5-7B-32k	41.40	9.9	63.5	34.20	23.20	53.00	55.30	40.07
SRA	43.20	14.5	65.1	35.80	25.10	53.72	55.96	41.91(+1.84)

5.3. Summarization Evaluation

We tested the enhancements brought by SRA on texts of different lengths using **LLaMA-3-8B-Instruct** and **LLaMA-2-13B-Chat**. Specifically, for **LLaMA-3-13B-Chat**, we started with texts of length 1000 tokens, collecting 100 cases at intervals of 500 tokens, up to a length of 4000 tokens. For **LLaMA-3-8B-Instruct**, we started with texts of length 2000 tokens, collecting 50 cases at intervals of 500 tokens, up to a length of 5000 tokens. We used GPT4o as the evaluation model following the GPT-4 evaluation protocol, comparing the outputs under SRA with the original model outputs. The results are illustrated in Figure 7, where we show the counts of “pure win” and “tie” cases. Here, the “pure win” refers to the winning number of SRA minus the winning number of the original model.

The results indicate that the benefits of SRA become increasingly evident as text length grows, with a declining number of ties and a steadily rising count of “pure wins”. Beyond a context length of 3000 for **LLaMA-2-13B-Chat** and 3500 for **LLaMA-3-8B-Instruct**, over half of the total samples show improvements compared to the original results when SRA is applied.

5.4. Results on Open-Source Benchmarks

Starting with LongBench, we selected 7 tasks including MultiFieldQA-EN (MFQA-EN), VCSUM (Wu et al., 2023), TREC (Li & Roth, 2002), SAMSum (Gliwa et al., 2019), LSHT (NLPCC, 2014), LCC (Guo et al., 2023b), and RepoBench-P (Liu et al., 2024c). The following results are illustrated in Table 1. Our SRA technique enables an

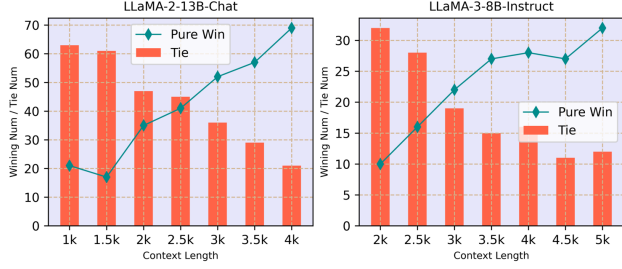


Figure 7. **Xsum** results on **LLaMA-2-13B-Chat** and **LLaMA-3-8B-Instruct**. SRA has demonstrated significant advantages in long-text comprehension and summarization, with these benefits becoming increasingly pronounced as the text length grows.

Table 2. **InfiniteBench Results of YaRN-Mistral**. Here, *Retrieve* encompasses both *Retrieve.PassKey* and *Retrieve.Number*. SRA notably enhances the retrieval capabilities of models trained with YaRN.

	Retrieve	En.Sum	En.MC
YaRN-Mistral	74.66	9.09	27.95
SRA	81.24	12.08	37.25

overall performance gain exceeding 1.8 across all models.

We further explored the performance of SRA on **YaRN-Mistral** within the InfiniteBench benchmark, with the results shown in Table 2. With the enhancements provided by SRA, the decline in retrieval and comprehension capabilities caused by PI modifications introduced by YaRN was significantly mitigated. This improvement further unlocks the potential of methods like YaRN and other PI approaches in the application of LLMs.

Finally, we conducted tests on the newly released LongBench v2, which includes a series of models using RoPE as their positional encoding method. The results are presented in Table 3. The results demonstrate that SRA exhibits excellent generalizability for LLMs utilizing RoPE as their positional encoding, significantly enhancing comprehension capabilities while maintaining high compatibility with CoT prompt engineering. For smaller models, the improvements brought by SRA are particularly pronounced, with most tasks on **Llama-3.1-8B-Instruct** achieving gains of over 2%. For larger LLMs, the most notable improvements are observed in long-text processing, with performance increases exceeding 2% in certain tasks. This is likely because advanced LLMs are already well-optimized for handling shorter contexts effectively. Moreover, regardless of task difficulty, the enhancements achieved through SRA remain consistent, demonstrating the robustness and reliability of our method.

5.5. Ablation Studies

In SRA, both the inter loop and outer loop are integral to its effectiveness. Even without scaling—where s_{in} and s_{ou} are set to 1—the basic “ReAttention” mechanism reduces perplexity during inference, as demonstrated in Table 4. Furthermore, extensive experiments reveal that the inter loop and outer loop enhance distinct aspects of model comprehension, as illustrated in Table 5. The inter loop primarily bolsters overall comprehension, making it particularly effective for tasks such as dialogue, summarization, and document understanding. In contrast, the outer loop excels in improving retrieval capabilities, especially for questions or keywords positioned near the end of prompts. Combining all of the components, SRA finally renders its superior effects.

5.6. Discussion of SRA Configurations

In our experiments, we tested numerous sets of SRA parameters to validate the gains brought by SRA. Despite variations in models and tasks, we derived a generalizable approach for tuning SRA parameters. More discussions and their corresponding experiments are exhibited in the Appendix ??.

From a model perspective, training methods and following tasks influence a model’s sensitivity to SRA. For example, while both are based on LLaMA, the LongChat series demonstrates greater sensitivity to SRA compared to the LLaMA series. Generally, models trained for retrieval tasks require a lower elimination threshold and smaller scale factors. Excessive values for these parameters can lead to incorrect outputs even when the correct position is identified, such as retrieving the correct context but returning an incorrect number in retrieval tasks. For models trained under standard conditions, larger parameter values are needed, particularly for the scaling factor, which directly affects the enhancement achieved.

In terms of context length, longer texts generally require smaller scaling factors. This is because the robustness of LLMs is limited, and excessively large scaling factors can impair the model’s language capabilities. Specifically, this manifests as fragmented sentences and incoherent expressions. While some keywords may still appear, they fail to form continuous and meaningful statements.

From a task perspective, as discussed in Sec. 5.5, the type of task significantly influences the configuration of the inter loop and outer loop scaling factors. For tasks such as QA and summarization, larger s_{in} and smaller s_{ou} are recommended. In contrast, for retrieval-based tasks, focusing the gains from SRA on the keywords in the final question—by setting a larger s_{ou} —yields better results.

Table 3. **SRA evaluation results (%) on LongBench v2.** Results under CoT prompting are highlighted with a gray background. SRA exhibits robust compatibility and enhancement effects for models employing RoPE as the positional encoding method, especially when integrated with CoT reasoning.

Model	Overall		Difficulty				Length (<32k; 32k-128k; >128k)					
			Easy		Hard		Short		Medium		Long	
Llama-3.1-8B-Instruct	30.0	30.4	30.7	36.5	29.6	26.7	35.0	34.4	27.9	31.6	25.9	21.3
SRA	31.3	32.0	31.9	38.2	30.7	28.5	37.3	36.9	29.3	33.1	27.2	23.4
Llama-3.3-70B-Instruct	29.8	36.2	34.4	38.0	27.0	35.0	36.7	45.0	27.0	33.0	24.1	27.8
SRA	31.0	37.2	35.1	39.2	27.9	35.8	37.5	48.1	28.6	34.3	25.8	28.4
Qwen2.5-72B-Instruct	39.4	38.8	43.8	42.2	36.7	36.7	44.4	50.0	34.0	28.8	41.7	39.8
SRA	41.2	41.4	44.5	43.6	36.9	39.1	45.2	51.3	35.9	31.4	43.5	41.6
Mistral-Large-Instruct-2407	26.6	33.6	29.7	34.4	24.8	33.1	37.8	41.1	19.5	31.2	22.2	25.9
SRA	28.0	34.6	30.5	36.2	26.3	34.5	39.1	43.4	20.2	31.5	24.3	27.2

Table 4. **Ablation study of LLaMA-3-8B on WikiText.** We use the last 200 words to calculate the perplexity.

Word Counts	1024	1536	2048
Original	5.58	5.61	5.22
+inter loop	5.57	5.60	5.21
+outer loop	5.57	5.59	5.20
RA	5.56	5.57	5.19

Table 5. **Ablation study of LLaMA-3-8B-Instruct on downstream tasks.**

Tasks	LongChat	MFQA-EN	LSHT
Original	59.5	41.50	24.25
+inter loop	62.5	42.35	26.80
+outer loop	68.9	41.78	25.20
SRA	70.0	42.71	28.00

6. Limitations

Variability of SRA Although we have established a relatively general set of guidelines, a small amount of task-specific testing remains unavoidable. This is particularly important for adjusting SRA parameters to suit different tasks. However, our experiments reveal that models within the same series generally exhibit similar characteristics, reducing the need for extensive testing to some extent.

Excessive application of SRA can cause significant disruptions to LLMs, potentially rendering them non-functional. Under normal use, SRA is characterized by a slight increase in perplexity compared to the original model. While some negative effects are present, the positive outcomes far outweigh them. From a task perspective, this slight increase in perplexity under normal usage has no impact on the quality or accuracy of the generated content.

Table 6. **Inference Speed of LLaMA 7B.** We report the running memory (denoted as ‘RM’) and speed in NVIDIA A100-80G.

Method	RM	Token/s
Normal	14.4 G	69.2
Flash Attention	13.7 G	77.9
SRA	14.6 G	64.8

Inference Efficiency A notable limitation of SRA is its reliance on explicit manipulation of attention scores after Softmax, as operations before the Softmax stage would disrupt the original distribution and introduce significant interference. This explicit computation prevents the use of certain attention acceleration techniques, such as FlashAttention (Dao et al., 2022), in conjunction with SRA, leading to slower inference speeds. However, the impact of SRA’s operations on pure processing time is minimal, as we only enhance a small fraction of tokens in heads with Hidden Gems and just in the prefilling stage. Comparisons are shown in Table 6. Moreover, the performance gains provided by SRA without additional training fully compensate for the impact of the extra time.

7. Conclusion

In this paper, we introduce SRA, a training-free method designed to enhance the contextual understanding capabilities of large language models. SRA achieves this by manually adjusting attention scores, amplifying the scores projected onto Hidden Gems, and trading off some model stability to improve retrieval and comprehension abilities. Through extensive experiments, we demonstrate the effectiveness of SRA across a variety of tasks, achieving significant performance improvements in retrieval and summarization tasks. Furthermore, SRA delivers notable enhancements even in open-ended long-text scenarios.

Impact Statement

Our goal is to enhance large language models’ reading comprehension and information retrieval capabilities without requiring any additional training. Our research is strongly oriented toward the industry, where cost is a crucial factor. Unlike previous research-focused work that requires significant resource investment to boost performance, our study emphasizes lightweight industrial implementation and practical deployment. In our view, our research makes an outstanding contribution.

The application scenarios for our research are highly extensive, as most large language models today are based on RoPE for positional encoding. Through a series of experiments, we have demonstrated the universality of our method. For instance, it can be applied to everyday tasks such as document summarization, inductive reasoning, long-text keyword retrieval, and memory in long and complex conversations, covering nearly all daily scenarios that require handling long texts.

For this work, the key point we need to emphasize remains the same: **no additional training is required**. Compared to the hundreds or thousands of A100 hours typically needed for training or fine-tuning, achieving immediate performance improvement through a plug-and-play method is exceptionally valuable.

References

- Ainslie, J., Lee-Thorp, J., de Jong, M., Zemlyanskiy, Y., Lebron, F., and Sanghai, S. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4895–4901. Association for Computational Linguistics, December 2023.
- Anthropic. Introducing the next generation of Claude. <https://www.anthropic.com/news/claude-3-family>, 2024. [Accessed 28-05-2024].
- Author, N. N. Suppressed for anonymity, 2021.
- Bai, J., Bai, S., Chu, Y., Cui, Z., Dang, K., Deng, X., Fan, Y., Ge, W., Han, Y., Huang, F., et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023a.
- Bai, Y., Lv, X., Zhang, J., Lyu, H., Tang, J., Huang, Z., Du, Z., Liu, X., Zeng, A., Hou, L., Dong, Y., Tang, J., and Li, J. Longbench: A bilingual, multitask benchmark for long context understanding, 2023b.
- Bai, Y., Tu, S., Zhang, J., Peng, H., Wang, X., Lv, X., Cao, S., Xu, J., Hou, L., Dong, Y., et al. Longbench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. *arXiv preprint arXiv:2412.15204*, 2024.
- Bondarenko, Y., Nagel, M., and Blankevoort, T. Quantizable transformers: Removing outliers by helping attention heads do nothing. *Advances in Neural Information Processing Systems*, 36:75067–75096, 2023.
- Chen, S., Wong, S., Chen, L., and Tian, Y. Extending context window of large language models via positional interpolation. *arXiv preprint arXiv:2306.15595*, 2023.
- Chiang, W.-L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J. E., Stoica, I., and Xing, E. P. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023.
- Dao, T., Fu, D., Ermon, S., Rudra, A., and Ré, C. Flashattention: Fast and memory-efficient exact attention with io-awareness. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 16344–16359, 2022.
- Dasigi, P., Lo, K., Beltagy, I., Cohan, A., Smith, N. A., and Gardner, M. A dataset of information-seeking questions and answers anchored in research papers, 2021.
- DeepSeek-AI, Liu, A., Feng, B., Wang, B., Wang, B., Liu, B., Zhao, C., Deng, C., Ruan, C., Dai, D., Guo, D., Yang, D., Chen, D., Ji, D., Li, E., Lin, F., Luo, F., Hao, G., Chen, G., Li, G., Zhang, H., Xu, H., Yang, H., Zhang, H., Ding, H., Xin, H., Gao, H., Li, H., Qu, H., Cai, J. L., Liang, J., Guo, J., Ni, J., Li, J., Chen, J., Yuan, J., Qiu, J., Song, J., Dong, K., Gao, K., Guan, K., Wang, L., Zhang, L., Xu, L., Xia, L., Zhao, L., Zhang, L., Li, M., Wang, M., Zhang, M., Zhang, M., Tang, M., Li, M., Tian, N., Huang, P., Wang, P., Zhang, P., Zhu, Q., Chen, Q., Du, Q., Chen, R. J., Jin, R. L., Ge, R., Pan, R., Xu, R., Chen, R., Li, S. S., Lu, S., Zhou, S., Chen, S., Wu, S., Ye, S., Ma, S., Wang, S., Zhou, S., Yu, S., Zhou, S., Zheng, S., Wang, T., Pei, T., Yuan, T., Sun, T., Xiao, W. L., Zeng, W., An, W., Liu, W., Liang, W., Gao, W., Zhang, W., Li, X. Q., Jin, X., Wang, X., Bi, X., Liu, X., Wang, X., Shen, X., Chen, X., Chen, X., Nie, X., Sun, X., Wang, X., Liu, X., Xie, X., Yu, X., Song, X., Zhou, X., Yang, X., Lu, X., Su, X., Wu, Y., Li, Y. K., Wei, Y. X., Zhu, Y. X., Xu, Y., Huang, Y., Li, Y., Zhao, Y., Sun, Y., Li, Y., Wang, Y., Zheng, Y., Zhang, Y., Xiong, Y., Zhao, Y., He, Y., Tang, Y., Piao, Y., Dong, Y., Tan, Y., Liu, Y., Wang, Y., Guo, Y., Zhu, Y., Wang, Y., Zou, Y., Zha, Y., Ma, Y., Yan, Y., You, Y., Liu, Y., Ren, Z. Z., Ren, Z., Sha, Z., Fu, Z., Huang, Z., Zhang, Z., Xie, Z., Hao, Z., Shao, Z., Wen, Z., Xu, Z., Zhang, Z., Li, Z., Wang, Z., Gu, Z., Li, Z., and Xie, Z. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model, 2024.

- Dettmers, T., Lewis, M., Belkada, Y., and Zettlemoyer, L. Llm.int8(): 8-bit matrix multiplication for transformers at scale, 2022.
- Duda, R. O., Hart, P. E., and Stork, D. G. *Pattern Classification*. John Wiley and Sons, 2nd edition, 2000.
- Emozilla. Dynamically Scaled RoPE further increases performance of long context LLaMA with zero fine-tuning, 2023. URL https://www.reddit.com/r/LocalLLaMA/comments/14mrgpr/dynamically_scaled_rope_further_increases/.
- Frantar, E. and Alistarh, D. Marlin: a fast 4-bit inference kernel for medium batchsizes. <https://github.com/IST-DASLab/marlin>, 2024.
- Ge, S., Zhang, Y., Liu, L., Zhang, M., Han, J., and Gao, J. Model tells you what to discard: Adaptive KV cache compression for LLMs. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.
- Gliwa, B., Mochol, I., Biesek, M., and Wawer, A. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pp. 70–79, Hong Kong, China, November 2019.
- Guo, C., Tang, J., Hu, W., Leng, J., Zhang, C., Yang, F., Liu, Y., Guo, M., and Zhu, Y. Olive: Accelerating large language models via hardware-friendly outlier-victim pair quantization. In *Proceedings of the 50th Annual International Symposium on Computer Architecture, ISCA '23*. ACM, June 2023a. doi: 10.1145/3579371.3589038. URL <http://dx.doi.org/10.1145/3579371.3589038>.
- Guo, D., Xu, C., Duan, N., Yin, J., and McAuley, J. Long-coder: A long-range pre-trained language model for code completion. In *International Conference on Machine Learning*, pp. 12098–12107. PMLR, 2023b.
- Han, I., Jayaram, R., Karbasi, A., Mirrokni, V., Woodruff, D., and Zandieh, A. Hyperattention: Long-context attention in near-linear time. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=Eh00d2BJIM>.
- Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., de Las Casas, D., and Hendricks. Training compute-optimal large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, 2024.
- Hsieh, C.-P., Sun, S., Krizan, S., Acharya, S., Rekesh, D., Jia, F., and Ginsburg, B. RULER: What’s the real context size of your long-context language models? In *First Conference on Language Modeling*, 2024.
- Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.
- Huang, L., Cao, S., Parulian, N., Ji, H., and Wang, L. Efficient attentions for long document summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1419–1436, Online, June 2021. Association for Computational Linguistics.
- Izacard, G., Lewis, P., Lomeli, M., Hosseini, L., Petroni, F., Schick, T., Dwivedi-Yu, J., Joulin, A., Riedel, S., and Grave, E. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24(251):1–43, 2023.
- Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. d. l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.
- Jiang, Z., Gao, L., Wang, Z., Araki, J., Ding, H., Callan, J., and Neubig, G. Retrieval as attention: End-to-end learning of retrieval and reading within a single transformer. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 2336–2349, December 2022.
- Joshi, M., Choi, E., Weld, D., and Zettlemoyer, L. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In Barzilay, R. and Kan, M.-Y. (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics.
- Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., and Yih, W.-t. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6769–6781, November 2020.
- Kazemnejad, A., Padhi, I., Natesan Ramamurthy, K., Das, P., and Reddy, S. The impact of positional encoding on length generalization in transformers. *Advances in Neural Information Processing Systems*, 36, 2024.
- Kearns, M. J. *Computational Complexity of Machine Learning*. PhD thesis, Department of Computer Science, Harvard University, 1989.
- Kočiský, T., Schwarz, J., Blunsom, P., Dyer, C., Hermann, K. M., Melis, G., and Grefenstette, E. 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>.
- Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J. E., Zhang, H., and Stoica, I. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.
- Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000.
- Li, D., Shao, R., Xie, A., Sheng, Y., Zheng, L., Gonzalez, J., Stoica, I., Ma, X., and Zhang, H. How long can context length of open-source LLMs truly promise? In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*, 2023.
- Li, X. and Roth, D. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*, 2002.
- Lin, J., Tang, J., Tang, H., Yang, S., Dang, X., Gan, C., and Han, S. Awq: Activation-aware weight quantization for llm compression and acceleration, 2023.
- Lin, J., Tang, J., Tang, H., Yang, S., Chen, W.-M., Wang, W.-C., Xiao, G., Dang, X., Gan, C., and Han, S. Awq: Activation-aware weight quantization for llm compression and acceleration, 2024.
- Liu, A., Liu, J., Pan, Z., He, Y., Haffari, G., and Zhuang, B. Minicache: KV cache compression in depth dimension for large language models. *arXiv preprint arXiv:2405.14366*, 2024a.
- Liu, H., Yan, W., Zaharia, M., and Abbeel, P. World model on million-length video and language with blockwise ringattention, 2024b.
- Liu, N. F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., and Liang, P. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024c.
- Liu, X., Yan, H., An, C., Qiu, X., and Lin, D. Scaling laws of roPE-based extrapolation. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024d. URL <https://openreview.net/forum?id=JO7k0SJ5V6>.
- Liu, Z., Wang, J., Dao, T., Zhou, T., Yuan, B., Song, Z., and Shrivastava. Deja vu: Contextual sparsity for efficient llms at inference time. In *International Conference on Machine Learning (ICML)*, pp. 22137–22176. PMLR, 2023.
- Michalski, R. S., Carbonell, J. G., and Mitchell, T. M. (eds.). *Machine Learning: An Artificial Intelligence Approach, Vol. I*. Tioga, Palo Alto, CA, 1983.
- Mitchell, T. M. The need for biases in learning generalizations. Technical report, Computer Science Department, Rutgers University, New Brunswick, MA, 1980.
- Mohtashami, A. and Jaggi, M. Random-access infinite context length for transformers. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, 2024.
- Narayan, S., Cohen, S. B., and Lapata, M. Don’t give me the details, just the summary! Topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1797–1807, October–November 2018.
- Newell, A. and Rosenbloom, P. S. Mechanisms of skill acquisition and the law of practice. In Anderson, J. R. (ed.), *Cognitive Skills and Their Acquisition*, chapter 1, pp. 1–51. Lawrence Erlbaum Associates, Inc., Hillsdale, NJ, 1981.
- Ni, J., Hernandez Abrego, G., Constant, N., Ma, J., Hall, K., Cer, D., and Yang, Y. Sentence-T5: Scalable sentence encoders from pre-trained text-to-text models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 1864–1874, Dublin, Ireland, May 2022.
- NLPCC. Task Definition for Large Scale Text Categorization at NLPCC 2014, 2014.
- NVIDIA. Nvidia ada lovelace professional gpu architecture. https://images.nvidia.com/aem-dam/en-zz/Solutions/technologies/NVIDIA-ADA-GPU-PROVIZ-Architecture-Whitepaper_1.1.1.pdf, 2023. [Accessed 28-05-2024].
- NVIDIA. Nvbench: Nvidia’s benchmarking tool for gpus, 2024. Available online: <https://github.com/NVIDIA/nvbench>.
- Nye, M., Andreassen, A. J., Gur-Ari, G., Michalewski, H., Austin, J., Bieber, D., Dohan, D., Lewkowycz, A., Bosma, M., Luan, D., Sutton, C., and Odena, A. Show your work: Scratchpads for intermediate computation with language models. In *Deep Learning for Code Workshop*, 2022.
- OpenAI. New models and developer products announced at devday. <https://openai.com/blog/new-models-and-developer-products-announced-at-devday#OpenAI>, November 2023. Accessed: 2024-01-31.

- OpenAI. Introducing GPT-4o: our fastest and most affordable flagship model. <https://platform.openai.com/docs/models>, 2024. [Accessed 28-05-2024].
- Oren, M., Hassid, M., Yarden, N., Adi, Y., and Schwartz, R. Transformers are multi-state RNNs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 18724–18741. Association for Computational Linguistics, November 2024.
- Peng, B., Quesnelle, J., Fan, H., and Shippole, E. Yarn: Efficient context window extension of large language models, 2023.
- Press, O., Smith, N., and Lewis, M. Train short, test long: Attention with linear biases enables input length extrapolation. In *International Conference on Learning Representations*, 2022.
- Rae, J. W., Potapenko, A., Jayakumar, S. M., Hillier, C., and Lillicrap, T. P. Compressive transformers for long-range sequence modelling. *arXiv preprint*, 2019. URL <https://arxiv.org/abs/1911.05507>.
- Ribar, L., Chelombiev, I., Hudlass-Galley, L., Blake, C., Luschi, C., and Orr, D. Sparq attention: bandwidth-efficient llm inference. In *ICML’24: Proceedings of the 41st International Conference on Machine Learning*, 2025.
- Samuel, A. L. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3(3):211–229, 1959.
- Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., and Liu, Y. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.
- Tang, J., Zhao, Y., Zhu, K., Xiao, G., Kasikci, B., and Han, S. QUEST: Query-aware sparsity for efficient long-context LLM inference. In *Forty-first International Conference on Machine Learning*, 2024.
- Thakkar, V., Ramani, P., Cecka, C., Shivam, A., Lu, H., Yan, E., Kosaian, J., Hoemmen, M., Wu, H., Kerr, A., Nicely, M., Merrill, D., Blasig, D., Qiao, F., Majcher, P., Springer, P., Hohnerbach, M., Wang, J., and Gupta, M. CUTLASS, January 2023. URL <https://github.com/NVIDIA/cutlass>.
- Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. Llama: Open and efficient foundation language models, 2023.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 6000–6010, 2017.
- Wang, S., Li, B. Z., Khabsa, M., Fang, H., and Ma, H. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.
- Wang, Y., Kordi, Y., Mishra, S., Liu, A., Smith, N. A., Khashabi, D., and Hajishirzi, H. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13484–13508, Toronto, Canada, July 2023.
- Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35: 24824–24837, 2022.
- Wu, H., Zhan, M., Tan, H., Hou, Z., Liang, D., and Song, L. VCSUM: A versatile Chinese meeting summarization dataset. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 6065–6079, Toronto, Canada, July 2023.
- Xiao, G., Lin, J., Seznec, M., Wu, H., Demouth, J., and Han, S. Smoothquant: Accurate and efficient post-training quantization for large language models, 2023.
- Xiao, G., Tian, Y., Chen, B., Han, S., and Lewis, M. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations (ICLR)*, 2024.
- Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W. W., Salakhutdinov, R., and Manning, C. D. Hotpotqa: A dataset for diverse, explainable multi-hop question answering, 2018.
- Ye, Z., Lai, R., Lu, R., Lin, C.-Y., Zheng, S., Chen, L., Chen, T., and Ceze, L. Cascade inference: Memory bandwidth efficient shared prefix batch decoding. <https://flashinfer.ai/2024/01/08/cascade-inference.html>, Jan 2024. URL <https://flashinfer.ai/2024/01/08/cascade-inference.html>. Accessed on 2024-02-01.
- Zandieh, A., Han, I., Daliri, M., and Karbasi, A. Kdeformer: Accelerating transformers via kernel density estimation. In *International Conference on Machine Learning (ICML)*, pp. 40605–40623. PMLR, 2023.
- Zhang, J., Lei, Y.-K., Zhang, Z., Han, X., Li, M., Yang, L., Yang, Y. I., and Gao, Y. Q. Deep reinforcement learning of transition states. *Physical Chemistry Chemical Physics*, 23(11):6888–6895, 2021.

- Zhang, J., Naruse, A., Li, X., and Wang, Y. Parallel top-k algorithms on gpu: A comprehensive study and new methods. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, New York, NY, USA, 2023a.
- Zhang, X., Chen, Y., Hu, S., Xu, Z., Chen, J., Hao, M., Han, X., Thai, Z., Wang, S., Liu, Z., et al. Infinitebench: Extending long context evaluation beyond 100k tokens. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15262–15277, 2024.
- Zhang, Z., Sheng, Y., Zhou, T., Chen, T., Zheng, L., Cai, R., Song, Z., Tian, Y., Ré, C., Barrett, C., Wang, Z. A., and Chen, B. H2o: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 34661–34710, 2023b.
- Zhao, Y., Lin, C.-Y., Zhu, K., Ye, Z., Chen, L., Zheng, S., Ceze, L., Krishnamurthy, A., Chen, T., and Kasikci, B. Atom: Low-bit quantization for efficient and accurate llm serving, 2024.
- Zheng, L., Li, Z., Zhang, H., Zhuang, Y., Chen, Z., Huang, Y., Wang, Y., Xu, Y., Zhuo, D., Xing, E. P., Gonzalez, J. E., and Stoica, I. Alpa: Automating inter- and Intra-Operator parallelism for distributed deep learning. In *16th USENIX Symposium on Operating Systems Design and Implementation (OSDI 22)*, pp. 559–578, Carlsbad, CA, Jul. 2022. ISBN 978-1-939133-28-1. URL <https://www.usenix.org/conference/osdi22/presentation/zheng-lianmin>.