# Attention Basin: Why Contextual Position Matters in Large Language Models

**Zihao Yi**[1], **Zhenqing Ling**[1,†], **Delong Zeng**[1,†], **Haohao Luo**[1], **Zhe Xu**[1], **Wei Liu**[2], **Jian Luan**[2], **Wanxia Cao**[2], **Ying Shen**[1,3,4,*]

[1]Sun Yat-Sen University, [2]MiLM Plus, Xiaomi Inc., [3]Peng Cheng Laboratory, [4]Guangdong Provincial Key Laboratory of Fire Science and Intelligent Emergency Technology

## Abstract

The performance of Large Language Models (LLMs) is significantly sensitive to the contextual position of information in the input. To investigate the mechanism behind this positional bias, our extensive experiments reveal a consistent phenomenon we term the *attention basin*: when presented with a sequence of structured items (e.g., retrieved documents or few-shot examples), models systematically assign higher attention to the items at the beginning and end of the sequence, while neglecting those in the middle. Crucially, our analysis further reveals that allocating higher attention to critical information is key to enhancing model performance. Based on these insights, we introduce Attention-Driven Reranking (**AttnRank**), a two-stage framework that (i) estimates a model's intrinsic positional attention preferences using a small calibration set, and (ii) reorders retrieved documents or few-shot examples to align the most salient content with these high-attention positions. **AttnRank** is a model-agnostic, training-free, and plug-and-play method with minimal computational overhead. Experiments on multi-hop QA and few-shot in-context learning tasks demonstrate that **AttnRank** achieves substantial improvements across 10 large language models of varying architectures and scales, without modifying model parameters or training procedures.

## 1 Introduction

Large Language Models (LLMs) have acquired vast amounts of knowledge from large-scale pretraining corpora and demonstrated exceptional capabilities in understanding and generating natural language. As a result, they have achieved remarkable success across a wide range of language tasks, from text summarization to multi-turn dialogue (Yi et al., 2024; Zhang et al., 2025b; Ling et al., 2025; Jiao et al., 2025). With the growing ability of LLMs to process increasingly long input sequences, Retrieval-Augmented Generation (RAG) has emerged as an effective paradigm for expanding the knowledge boundaries of LLMs and improving answer accuracy. By providing relevant external documents at inference time, RAG enables models to perform competitively even in complex scenarios such as multi-hop question answering and few-shot learning, where the input naturally consists of multiple semantically independent documents or examples organized into a structured context block (Li et al., 2024; Yi et al., 2025).

However, behind this apparent success lies a fundamental vulnerability: the performance of LLMs is highly sensitive to the position of information within the input context (Liu et al., 2024). This positional bias poses a critical bottleneck—models often fail to *utilize* long contexts effectively. Even when all necessary information is present, performance can degrade significantly if key content is placed in regions of low attention, leading to suboptimal and unpredictable outcomes (Xiao et al., 2024; Jiao et al., 2024; Li et al., 2022).

This vulnerability manifests empirically as the "lost-in-the-middle" (LIM) phenomenon, where models show a clear preference for information at the beginning or end of the context (Liu et al., 2024). Although LIM provides an insightful phenomenological analysis, it describes the effect rather than elucidating the underlying cause. Concurrently, other mitigation strategies, such as fine-tuning for better instruction following (Li et al., 2023), explore different avenues but often incur substantial computational overhead without addressing the core mechanism of the bias.

To move beyond surface-level symptoms and uncover the root cause of positional bias, we turn our investigation to the very core of how LLMs process context: the attention mechanism (Vaswani et al., 2017). To this end, our empirical analysis

[†]Equal contribution.
[*]Correspondence to: sheny76@mail.sysu.edu.cn



*Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 26086–26106

in a meticulously designed experimental sandbox across 10 mainstream LLMs reveals a consistent and systematic pattern, which we term the *attention basin*: a U-shaped attention distribution over the *entire structured input block* (e.g., multiple retrieved documents or in-context examples), rather than an artifact of individual tokens or special positions. Crucially, we show that this pattern disappears once structural delimiters are removed, indicating that the attention basin is a *structural-level bias* rather than a token-level effect.

Armed with this mechanistic insight, we propose Attention-Driven Reranking (**AttnRank**), a principled framework that translates structural attention bias into an actionable inference-time strategy. **AttnRank** is not a heuristic reordering rule, but a mechanism-driven approach that explicitly aligns document relevance with a model's intrinsic attention profile. First, we probe the model's intrinsic attention landscape using a small, representative calibration set. Second, we leverage this map to re-rank the input context, strategically aligning the most critical information with the model's natural high-attention regions. Extensive evaluations on multi-hop QA and few-shot learning tasks show that **AttnRank** consistently improves performance across 10 mainstream LLMs across varying architectures and scales, achieving significant gains without any model training or parameter modification.

This work makes three key contributions:

- **Uncovering the Mechanism of Positional Bias:** We empirically and theoretically identify the *attention basin* phenomenon as a core mechanistic driver of positional bias: LLMs intrinsically allocate higher attention to the start and end of the overall structural block of context, aligning critical information with high-attention zones is crucial for effective context utilization.

- **Introducing Attention-Driven Reranking Method:** We propose a novel, lightweight and training-free method **AttnRank** that first maps a model's inherent positional attention preferences and then reorders the input to exploit these preferences for improved performance.

- **Validation of AttnRank's Effectiveness:** Extensive experiments on multi-hop QA and few-shot learning tasks demonstrate that **AttnRank** consistently outperforms baseline strategies across 10 mainstream LLMs of varying architectures and scales, effectively mitigating positional bias and significantly enhancing information utilization.

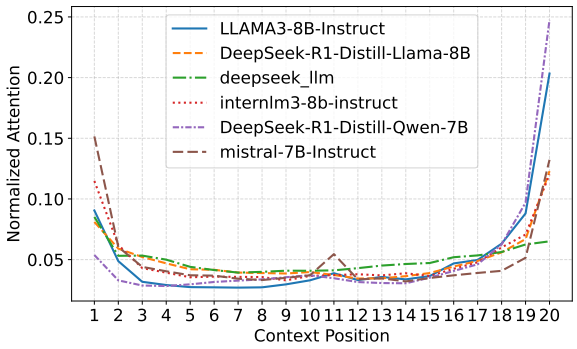
## 2 Related Works

Positional sensitivity is a well-known limitation of Large Language Models (LLMs), adversely affecting applications such as Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) and in-context learning (Brown et al., 2020). Existing studies mainly focus on characterizing and mitigating this positional bias.

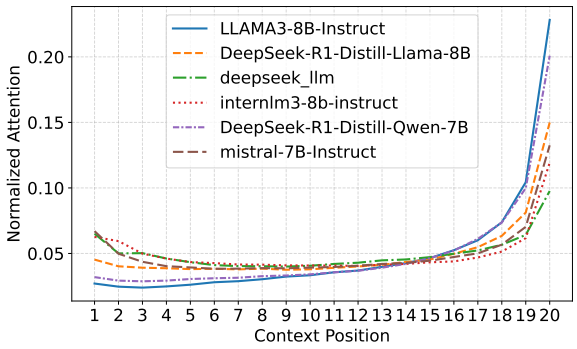
**Characterizing Positional Bias.** Initial research characterized positional bias through phenomena like the "Lost in the Middle" (LIM) effect, a U-shaped performance curve where models recall information from the context's edges more effectively than its center (Liu et al., 2024). However, such descriptions are phenomenological and do not elucidate the underlying mechanisms. Subsequent work has proposed mechanistic explanations, such as the "attention sink" on initial tokens (Xiao et al., 2024) or a lack of explicit supervision for positional invariance during training (An et al., 2024). Nevertheless, these theories remain incomplete: the attention sink only explains the primacy effect at a token level, and the supervision hypothesis lacks a concrete, quantifiable mechanism. Consequently, a fundamental gap persists in understanding and empirically evaluating positional bias.

**Mitigating Positional Bias.** Most mitigation methods rely on context reranking. Heuristic approaches, such as repositioning key documents (Liu et al., 2024), are often unstable, while LLM-based rerankers (Wang et al., 2025; Guo et al., 2025; Zhang et al., 2025a) incur substantial computational and latency costs. Training-based methods can reduce bias (An et al., 2024) but require expensive fine-tuning and may harm general model performance.

Overall, although prior work has identified positional bias and explored mitigation strategies, a significant gap remains between limited mechanistic understanding and practical, efficient solutions. This work bridges this gap by providing a deeper analysis of positional bias and proposing a lightweight, principled mitigation approach that avoids the drawbacks of existing methods.




(a) Attention Basin Phenomenon




(b) Effect of Disrupted Delimiters

Figure 1: (a) The model exhibits a U-shaped attention distribution, prioritizing tokens at the context's start and end boundaries. (b) This pattern disappears following the removal of structural delimiters, indicating that the phenomenon stems from the model's awareness of coherent segment boundaries.

## 3 Observation and Analysis

In this section, we present our key discoveries that illuminate the mechanisms behind the performance variations of Large Language Models (LLMs) due to contextual position. Our analysis unfolds in three parts: first, we identify a consistent, structure-aware attention pattern we term the *attention basin*; second, we establish a direct link between this attention pattern and model performance; and third, we pinpoint which layers are most critical in forming this positional preference.

### 3.1 The *Attention Basin* Phenomenon

Inputs to LLMs for tasks like RAG or few-shot learning often consist of multiple, distinct segments. We model such inputs as a sequence of structural blocks $S = \{t, d_1, \ldots, d_k, q\}$, where $t$ is a task-defining template, $d_i$ are semantically coherent blocks of content (e.g., retrieved documents), and $q$ is the user's query.

While prior work has observed elevated attention on initial tokens and local neighborhoods (Ma et al., 2024), attention distribution across document-level blocks remains underexplored. To study this, we identify the token ranges corresponding to each document block $D = d_1, \ldots, d_k$ in the attention matrix and compute the mean attention from query tokens to each block.

As shown in Figure 1a, this analysis reveals a consistent U-shaped pattern across LLMs, where documents near the beginning and end receive substantially higher attention. We refer to this phenomenon as the *attention basin*.

This discovery raises a critical question: Is the attention basin a simple artifact of absolute position (i.e., the model just likes the start and end of any text), or is it driven by the model's perception of the input's structure? To distinguish between these possibilities, we designed a disruption experiment aimed at understanding the root cause of this phenomenon. We systematically dismantled the input's structure by removing punctuation, capitalization, and explicit delimiters like "Document [1]", effectively blending the distinct documents into a single, unstructured block of text.

As shown in Figure 1b, the attention basin effect vanished entirely after the structure was removed. This result provides a profound insight: the phenomenon is not arbitrary. Instead, it reveals that the attention basin is a structural-level analogue to the well-known token-level primacy and recency effects. Just as models are known to over-attend to the first and last tokens of an entire sequence, our experiment demonstrates that they apply a similar heuristic at a higher level of abstraction, granting special status to the structural blocks positioned at the edges of the context. The model recognizes the collection of documents as a set and focuses its attention on the boundaries of that set.

### 3.2 How Attention Influences LLMs' Performance

The attention basin provides a mechanistic explanation for the widely observed lost-in-the-middle effect (Liu et al., 2024). If documents at the edges receive more attention, it follows that their content would more strongly influence the model's output. We hypothesize that a document's contribution to the final answer is directly proportional to the attention it receives.

To formalize this intuition, we analyze the rela-

tionship between a document's attention and the model's output probability. Let $\bar{\alpha}_d = \frac{1}{L}\sum_{l=1}^{L}\alpha_d^{(l)}$ be the cross-layer average attention weight allocated to a document $d$, and let $P(y^*|\cdot)$ be the generation probability of the correct answer $y^*$. Under the simplifying assumption of semi-orthogonal document representations (Assumption C.1), our theoretical analysis yields the following proposition, with the full proof in Appendix C.

**Proposition 3.1** (Attention-Probability Monotonicity)**.** *For a correct document $d^*$ and any other document $d_j$, the partial derivatives of the correct answer's probability $P(y^*|\cdot)$ with respect to their average attention weights satisfy:*

$$\frac{\partial P(y^*|\cdot)}{\partial \bar{\alpha}_{d^*}} > \left|\frac{\partial P(y^*|\cdot)}{\partial \bar{\alpha}_{d_j}}\right| \geq 0. \quad (1)$$

This proposition mathematically confirms that increasing attention on the correct document $d^*$ provides the most effective path to improving model performance. Corollary C.7 in the appendix further leverages this insight to explain the lost-in-the-middle phenomenon, demonstrating that placing crucial documents in high-attention regions (i.e., the edges) maximizes the probability of generating the correct answer.

We then validated this theory empirically using the HotpotQA dataset (Yang et al., 2018). We constructed inputs with two ground-truth documents ($d_1$, $d_2$) and one irrelevant noise document $n$ . We tested all six permutations of these documents, measuring both the attention distribution and the final QA accuracy for each. We categorized permutations into two groups: those where the ground-truth documents received the highest cumulative attention, and those where the noise document did.

The results in Figure 2 are unequivocal. Permutations where the correct documents received the most attention (blue bars) significantly outperformed those where the noise document was attended to most (red bars). Remarkably, this attention-optimized ordering not only mitigated the impact of the distractor but, in some cases, even surpassed the performance of the noise-free baseline (orange bar). This demonstrates that controlling positional attention is a powerful mechanism for improving model robustness and accuracy.

In addition, we find that when the input structure is disrupted, simply restoring a canonical (ascending/original) document order yields the best performance, further supporting our insight that, after structure breaking, a monotonically increasing attention pattern—and assigning higher attention to the correct documents—can improve robustness (see Appendix G).

### 3.3 The Critical Role of Shallow Attention Layers

Prior studies (Artzy and Schwartz, 2024; Van Aken et al., 2019) have shown that attention across different layers contribute unequally to its behavior. Inspired by Abnar et al. (Abnar and Zuidema, 2020), who demonstrate that information becomes increasingly entangled in deeper layers, we analyze this effect by decomposing the attention score at a given position $p$ and layer $l$. To formalize the *attention basin phenomenon*, our analysis is based on the assumption (detailed in Assumption C.2) that expected attention can be decomposed into two parts: a deterministic, position-dependent bias $f(p)$ and a content-driven, position-agnostic component $\epsilon_p^{(l)}$. This is expressed as: $\mathbb{E}[\alpha_p^{(l)}] = f(p) + \epsilon_p^{(l)}$.

To quantify the balance between these two forces, we define a ratio $\rho(l)$ of their variances: $\rho(l) = \mathbb{V}[\epsilon_p^{(l)}]/\mathbb{V}[f(p)]$. This ratio leads to the following hypothesis.

*Hypothesis* 3.2 (Layer-wise Attention Regimes). There exists a layer depth threshold $L^*$ that partitions the model into two regimes based on the attention-type variance ratio $\rho(l)$:

1. **Position-dominated regime** ($l < L^*$): $\rho(l) < 1$, where positional bias variance exceeds content variance.

2. **Content-dominated regime** ($l \geq L^*$): $\rho(l) \geq 1$, where content-based attention variance becomes dominant.

This hypothesis is derived from the empirical observation that content-based attention variance $\mathbb{V}[\epsilon_p^{(l)}]$ increases with layer depth as self-attention becomes more semantic (see Appendix C, Hypothesis C.8 for the full derivation and empirical verification).

If this hypothesis holds, **shallow-layer attention distributions more accurately reflect the model's structural and positional focus** on different input segments. This insight is crucial, as it suggests that the attention patterns from early layers, where the positional signal $f(p)$ is strongest, are the most reliable signal for understanding and manipulating the model's positional preferences. We empirically verify this hypothesis in the following experiment.

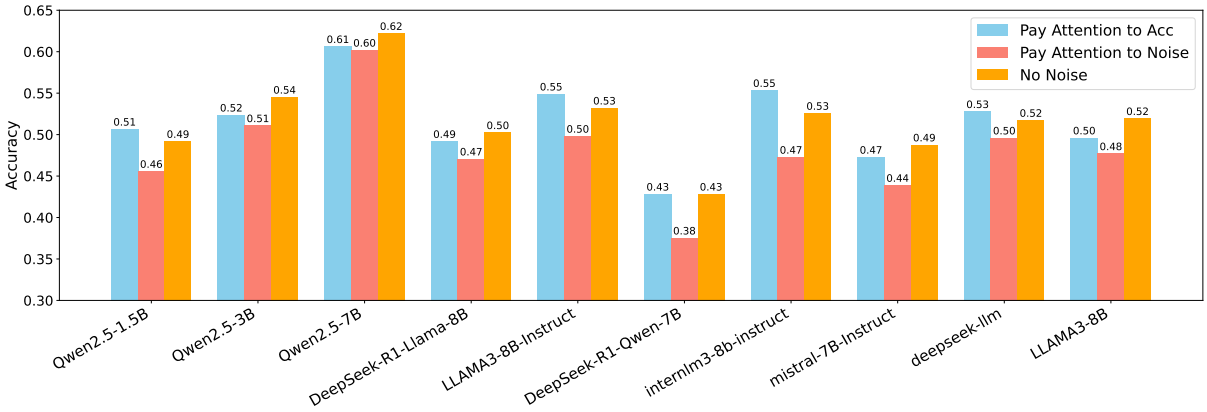



Figure 2: Model QA accuracy on HotpotQA across all permutations of two relevant documents and one noise document. Blue bars: permutations where relevant documents receive the highest attention. Red bars: permutations where the noise document receives the highest attention. Orange bar: noise-free baseline. Aligning relevant documents with high-attention positions consistently yields the best performance.

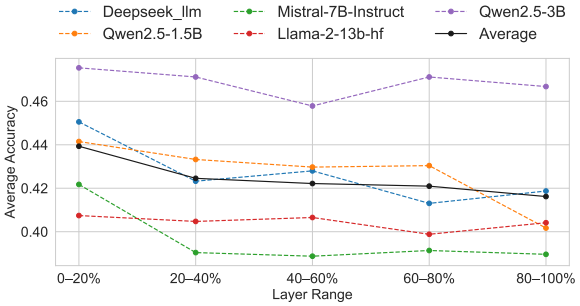



Figure 3: Accuracy of a reranking strategy based on attention from different Transformer layers. Reranking using shallow-layer attention consistently outperforms using deeper layers, indicating that LLM's core positional bias is established early.

To verify this, we designed a reranking experiment. For a given query, we retrieved five relevant documents. We then used the attention scores from different layers of the model to determine the optimal position for the most important document. As shown in Figure 3, reranking based on attention from the shallowest layers consistently yielded the highest QA accuracy. This confirms our hypothesis: the model's foundational positional bias is set early in the forward pass, making shallow-layer attention the most effective signal for understanding and ultimately mitigating this bias.

## 4 Methodology

As shown in Section 3, LLM performance is highly sensitive to document ordering due to an intrinsic *attention basin*. Rather than counteracting this bias, we propose to exploit it. We introduce the **Attention-Driven Reranker (AttnRank)**, a lightweight, training-free method that aligns document relevance with the model's inherent positional attention preferences. By placing the most important documents at positions where the model naturally focuses, **AttnRank** mitigates positional bias and improves performance. As illustrated in Figure 4, the framework consists of two stages: attention distribution extraction and attention-based reranking.

### 4.1 Step 1: Attention Distribution Extraction

The foundational step of **AttnRank** is to create a stable, general-purpose "attention profile" for a given LLM. This profile serves as a map of the model's positional biases. Based on our findings in Section 3, we use the shallowest attention layer, as it provides the purest signal of the model's intrinsic positional preference.

To generate this profile, we craft a set of probe inputs $S_i = \{t, d_1, \ldots, d_k, q\}$, where $t$ is a fixed task template, $D = \{d_1, \ldots, d_k\}$ are placeholder documents, and $q$ is a query. We then compute the average attention paid by the query tokens to each document position across multiple samples:

$$\begin{aligned} A &= \{a_1, \ldots, a_k\} \\ &= \frac{1}{N} \sum_{i=1}^{N} \text{Attention}(\{d_1, \ldots, d_k\}_i \mid S_i), \end{aligned} \tag{2}$$

where $A$ is the final attention profile, $a_j$ is the average attention score for the $j$-th position, $N$ is the number of probe samples, and $\text{Attention}(\cdot)$ extracts the mean attention from the query to each document block.

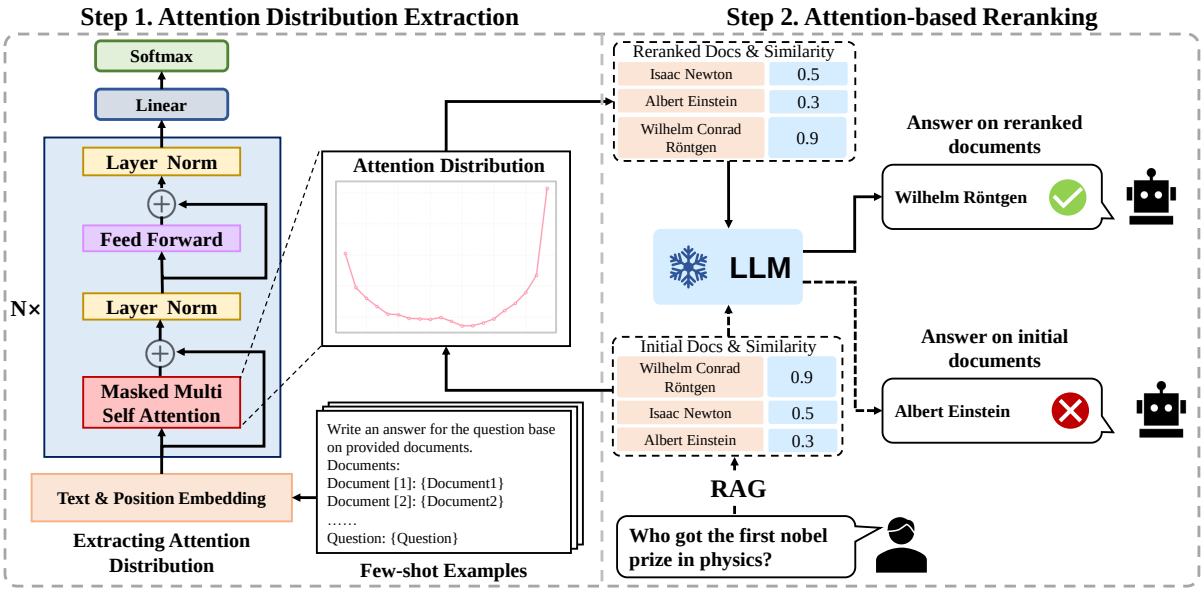



Figure 4: Overview of the AttnRank framework. Step 1: Profiling Positional Attention—We perform a one-time, low-cost analysis using **first-layer attention** to capture the model's intrinsic positional attention pattern (the attention basin) across document positions. Step 2: Attention-driven Reranking—For any new query, we reorder the retrieved documents, mapping the most relevant document (highest similarity) to the position with the highest profiled attention score, thus aligning relevance with the model's natural focus.

Crucially, we find that this profiling process is highly efficient. Our experiments show that a stable attention profile can be established with a remarkably small number of samples—often as few as 400. For some models, the characteristic *attention basin* pattern emerges with just a single example (see Appendix E). This one-time, low-cost profiling step yields a reusable attention map that captures the essence of the model's positional bias.

### 4.2 Step 2: Attention-based Reranking

Once the model's attention profile $A$ is established, it can be deployed to optimize document ordering for any subsequent task, such as Retrieval-Augmented Generation (RAG). The reranking procedure is as follows:

1. **Retrieve:** For a given user query, use a standard retriever to fetch the top-$k$ most relevant documents, $D = \{d_1, \ldots, d_k\}$, ranked by a similarity metric .

2. **Rerank:** We reorder the retrieved documents using the attention profile $A$, assigning higher-relevance documents to positions with higher attention scores (e.g., mapping $d_1$ to the highest-attention position, $d_2$ to the second-highest, and so on).

3. **Generate:** Concatenate the reordered documents with the query and prompt, and feed the final input into the LLM for generation.

By synchronizing the relevance hierarchy of the documents with the attention hierarchy of the model, **AttnRank** ensures that the most critical information is placed exactly where the model is hardwired to look. This alignment preemptively resolves the conflict between document importance and positional bias, allowing the model to focus its computational resources effectively and avoid distractions from less relevant information.

**AttnRank** is model-agnostic and incurs negligible overhead, requiring only a one-time profiling step. As an input-level preprocessing method, it is fully compatible with existing inference acceleration frameworks such as FlashAttention and vLLM, enabling seamless integration without modifying model parameters or inference pipelines.

## 5 Experiments

To validate the effectiveness of **AttnRank**, we raise the following three key Research Questions (**RQs**). We conduct comprehensive experiments on various datasets using 10 mainstream LLMs across different architectures and scales to systematically address these **RQs**:

| Models | Random | Descending | Ascending | LIM(Liu et al., 2024) | AttnRank (Ours) |
|---|---|---|---|---|---|
| InternLM3-8B-Instruct (2024) | 41.41 | **42.92** | 40.91 | 41.91 | 41.56 |
| Mistral-7B-Instruct (2023) | 39.47 | 38.65 | 38.90 | 41.52 | **42.17** |
| LLAMA-2-13B-hf (2023) | 38.37 | 38.20 | 40.74 | 39.67 | **41.00** |
| LLAMA3-8B (2024) | 41.36 | 40.76 | **43.19** | 42.09 | 43.09 |
| LLAMA3-8B-Instruct (2024) | 44.77 | 45.04 | 44.81 | 44.02 | **46.32** |
| DeepSeek-R1-Distill-Llama-8B (2025) | 39.45 | 38.59 | 39.81 | 39.82 | **41.77** |
| DeepSeek-LLM (2024) | 42.14 | 41.20 | 41.51 | 42.22 | **45.05** |
| Qwen 2.5 1.5B (2024) | 40.62 | 41.23 | 43.56 | 39.48 | **44.14** |
| Qwen 2.5 3B (2024) | 45.77 | 46.42 | 47.45 | 45.62 | **47.54** |
| Qwen 2.5 7B (2024) | 52.32 | 53.31 | **54.64** | 52.18 | 54.55 |
| **Average Accuracy** | 42.57 | 42.63 | 43.55 | 42.85 | **44.72** |

Table 1: Answer accuracy (%) on the HotpotQA dataset using different document ordering strategies. Five documents are retrieved per question, and models are evaluated on how accurately they answer the question based on the ordered input.

**RQ1:** Does placing informative documents in high-attention slots improve LLM's performance?
**RQ2:** How does the effectiveness of **AttnRank** generalize across a diverse range of LLM architectures and scales?
**RQ3:** Is **AttnRank** robust across different multi-document tasks and datasets?

**Baselines** Four reranking baselines are applied to the retrieved document set: (1) **Random**: Documents shuffled randomly. (2) **Similarity Descending**: Documents sorted by retrieval similarity in descending order. (3) **Similarity Ascending**: Documents sorted by retrieval similarity in ascending order. (4) **Lost-in-the-Middle (LIM)** (Liu et al., 2024): Places the highest-similarity documents at the beginning and end of the input sequence.

**Datasets** We evaluate **AttnRank** on four widely used benchmark datasets spanning diverse task types, input formats, and context lengths. For multi-hop question answering, we adopt **HotpotQA** (Yang et al., 2018) and **2WikiMultiHopQA** (Ho et al., 2020), both of which require reasoning across multiple documents and involve long retrieved contexts of approximately 8K tokens. To assess generalization to alternative input structures and shorter contexts, we additionally use the **MultiWOZ 2.1 and 2.4** (Eric et al., 2020; Ye et al., 2022) datasets in few-shot code generation and multi-domain dialogue settings. Compared to multi-hop QA, MultiWOZ contains shorter exemplars (around 600 tokens) and distinct query styles, offering a complementary evaluation setting.

**Metrics** We evaluate multi-hop QA performance using **answer average accuracy** as a straightforward metric. For few-shot experiment, we use the **Joint Goal Accuracy (JGA)** (Henderson et al., 2014) as the evaluation metric. For each turn in a dialogue, a dialogue state is considered correct only if it **exactly matches** the ground truth.

### 5.1 Multi-hop QA experiment

Multi-hop QA is an ideal testbed for our hypothesis, as it requires models to identify and integrate dispersed evidence across multiple documents, making performance strongly dependent on effective context utilization.

**Experimental setup** For each question, five candidate documents are retrieved by beam-retriever (Zhang et al., 2024). As shown in Appendix A), all reranked document lists are fed into a unified QA model with a identical prompt template.

**Experiment results and analyze** As shown in Tables 1 and 2, **AttnRank** consistently outperforms baselines on HotpotQA (44.72% vs. Random 42.57%, Descending 42.63%, Ascending 43.55%, LIM 42.85%) and 2WikiMultiHopQA (34.72% vs. Random 32.75%, Descending 32.85%, Ascending 34.50%, LIM 32.10%). Notably, while ascending order occasionally performs competitively, particularly on 2WikiMultiHopQA, **AttnRank** still consistently yields equal or superior accuracy, indicating that simple heuristics do not fully capture the model's attention dynamics. These results answer **RQ1** that placing the most informative documents in positions where the model's attention is most focused significantly enhances its ability to reason across multiple hops. The consistent improvements across different architectures (e.g., LLAMA3, Qwen, Mistral, DeepSeek) and model

| Models | Random | Descending | Ascending | LIM (Liu et al., 2024) | AttnRank (Ours) |
|---|---|---|---|---|---|
| InternLM3-8B-Instruct (2024) | 39.14 | **40.87** | 38.41 | 39.94 | 39.56 |
| Mistral-7B-Instruct (2023) | 29.64 | 28.47 | **33.14** | 27.74 | 32.54 |
| LLAMA-2-13B-hf (2023) | 30.21 | 30.27 | **31.79** | 30.18 | **31.79** |
| LLAMA3-8B (2024) | 32.17 | 30.78 | 33.39 | 30.35 | **33.97** |
| LLAMA3-8B-Instruct (2024) | 37.39 | 37.91 | **39.35** | 36.59 | 38.14 |
| DeepSeek-R1-Distill-Llama-8B (2025) | 27.09 | 26.68 | 28.02 | 26.78 | **28.24** |
| DeepSeek-LLM (2024) | 27.83 | 28.12 | 30.54 | 28.16 | **31.31** |
| Qwen 2.5 1.5B (2024) | 30.05 | 29.48 | 33.34 | 29.35 | **33.97** |
| Qwen 2.5 3B (2024) | 32.72 | **34.33** | 33.55 | 32.18 | 34.08 |
| Qwen 2.5 7B (2024) | 41.22 | 41.55 | 43.44 | 39.71 | **43.55** |
| **Average** | 32.75 | 32.85 | 34.50 | 32.10 | **34.72** |

Table 2: Answer accuracy (%) on 2WikiMultiHopQA using different document ordering strategies. Five documents are retrieved per question, and models are evaluated on answer correctness.

sizes ranging from 1.5B to 13B further validate the generalizability of **AttnRank**, thereby answering **RQ2**. Additional experimental details, including attention distribution and case studies, are provided in Appendix F.

### 5.2 Few-shot experiment

Few-shot generation task involves generating outputs based on a small number of example demonstrations and a user query. We design an experiment to assess the **AttnRank** in this setting. The IC-DST algorithm (Hu et al., 2022) retrieves a small set of in-context examples to generate SQL queries, thereby extracting user intent from dialogue history and maintaining an up-to-date dialogue state. The precision of the dialogue state is strongly correlated with the quality of the code generation, for which we choose to test the effectiveness of the **AttnRank** based on the IC-DST algorithm.

**Experimental setup** For each dialogue turn, the trained IC-DST retriever fetch $k$ example dialogues from the few-shot context pool. The same five reranking strategies described in Section 5.1 are applied to the retrieved examples. The sorted examples and the current dialogue history are then passed to the Code Llama (Roziere et al., 2023) with a fixed prompt template (see Appendix A).

**Experiment results and analyze** As shown in Table 3, **AttnRank** outperforms all baselines on MultiWOZ 2.1 and 2.4, improving over random by 1.58%. All structured ordering strategies yield gains over random, indicating that systematic example ordering enhances context relevance. **AttnRank** 's additional improvements over Ascending and LIM confirm that aligning high-value examples with the model's attention peaks further boosts extraction accuracy, validating our hypothesis.

| Method | MultiWOZ 2.1 | MultiWOZ 2.4 | Average |
|---|---|---|---|
| Random | 41.12 | 50.11 | 45.62 |
| Descending | 42.10 | 50.61 | 46.36 |
| Ascending | 42.67 | 51.16 | 46.92 |
| LIM(Liu et al., 2024) | 42.44 | 51.14 | 46.79 |
| **AttnRank (Ours)** | **42.89** | **51.51** | **47.20** |

Table 3: Joint Goal Accuracy (%) on MultiWOZ 2.1 and 2.4, under different context retrieval reranking methods.

Overall, the strong performance on both multi-hop QA and few-shot learning—spanning four datasets and two distinct task types—provides a comprehensive and positive answer to **RQ3**. Our findings demonstrate that **AttnRank** is a robust and effective method for mitigating negative positional effects across diverse scenarios.

## 6 Conclusion

In this work, we identified a fundamental mechanism governing positional bias in Large Language Models: the attention basin phenomenon. We demonstrated that LLMs exhibit a systematic tendency to focus on the beginning and end of a structured input block, a predictable pattern rather than a random flaw. This core insight allowed us to propose Attention-Driven Reranking (**AttnRank**), a lightweight and training-free framework that transforms this bias from a liability into an asset. By strategically reordering input items to align salient information with the model's natural attention peaks, **AttnRank** effectively enhances knowledge utilization. Crucially, as a model-agnostic and parameter-free method, it requires no architectural modifications and seamlessly integrates with existing pipelines. This makes it fully compatible

with modern acceleration frameworks, offering a rare combination of improved accuracy and high efficiency. We believe this principle of "attention alignment" opens new avenues for research, and we discuss potential limitations and future directions in Sec 7 to inspire further exploration.

## 7 Limitations

The proposed **AttnRank** method mitigates detrimental positional bias by aligning document relevance with the model's intrinsic attention patterns, leading to consistent performance gains. However, since most closed-source large language models do not expose attention scores through their APIs, the applicability of our approach to such models cannot be directly verified at present.

Moreover, our theoretical analysis relies on simplified assumptions (e.g., approximate independence of document representations and additive position–attention effects) to enable tractable reasoning. These assumptions do not strictly hold in real transformer models and therefore limit the interpretation of the theory as an exact description of model internals.

Finally, while our method exploits existing positional bias, an equally important direction is to *reduce* such bias. Possible strategies include **(i) active attention modulation**, where attention scores are adjusted at inference time to compensate for low-attention regions (e.g., enhancing middle-position attention), and **(ii) bias-aware training**, where models are fine-tuned on data that deliberately places key information in traditionally low-attention positions. Exploring such approaches remains an important direction for future work.

## Acknowledgments

This work was supported in part by the New Generation Artificial Intelligence-National Science and Technology Major Project (2025ZD0123003), the National Natural Science Foundation of China Enterprise Innovation and Development Joint Fund (Artificial Intelligence Field) Key Support Projects (U25B2072), and The Major Key Project of PCL (Grant No. PCL2025A17).

## References

Samira Abnar and Willem Zuidema. 2020. Quantifying attention flow in transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4190–4197.

Shengnan An, Zexiong Ma, Zeqi Lin, Nanning Zheng, Jian-Guang Lou, and Weizhu Chen. 2024. Make your llm fully utilize the context. *Advances in Neural Information Processing Systems*, 37:62160–62188.

Amit Ben Artzy and Roy Schwartz. 2024. Attend first, consolidate later: On the importance of attention in different LLM layers. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 177–184, Miami, Florida, US. Association for Computational Linguistics.

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, and 1 others. 2024. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, and 1 others. 2024. Internlm2 technical report. *arXiv preprint arXiv:2403.17297*.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Goyal, Peter Ku, and Dilek Hakkani-Tur. 2020. Multiwoz 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 422–428.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Minghao Guo, Qingcheng Zeng, Xujiang Zhao, Yanchi Liu, Wenchao Yu, Mengnan Du, Haifeng Chen, and Wei Cheng. 2025. DeepSieve: Information Sieving via LLM-as-a-Knowledge-Router. *arXiv e-prints*, arXiv:2507.22050.

Matthew Henderson, Blaise Thomson, and Steve Young. 2014. Word-based dialog state tracking with recurrent neural networks. In *Proceedings of the 15th annual meeting of the special interest group on discourse and dialogue (SIGDIAL)*, pages 292–299.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop

qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625.

Yushi Hu, Chia-Hsuan Lee, Tianbao Xie, Tao Yu, Noah A Smith, and Mari Ostendorf. 2022. In-context learning for few-shot dialogue state tracking. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2627–2643.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. *arXiv e-prints*, arXiv:2310.06825.

Qirui Jiao, Daoyuan Chen, Yilun Huang, Yaliang Li, and Ying Shen. 2024. From training-free to adaptive: Empirical insights into mllms' understanding of detection information. *arXiv preprint arXiv:2401.17981*.

Qirui Jiao, Daoyuan Chen, Yilun Huang, Xika Lin, Ying Shen, and Yaliang Li. 2025. Detailmaster: Can your text-to-image model handle long prompts? *arXiv preprint arXiv:2505.16915*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 611–626.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Dacheng Li, Rulin Shao, Anze Xie, Ying Sheng, Lianmin Zheng, Joseph Gonzalez, Ion Stoica, Xuezhe Ma, and Hao Zhang. 2023. How long can context length of open-source llms truly promise? In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*.

Ruosen Li, Zimu Wang, Son Tran, Lei Xia, and Xinya Du. 2024. Meqa: A benchmark for multi-hop event-centric question answering with explanations. *Advances in Neural Information Processing Systems*, 37:126835–126862.

Yinghui Li, Qingyu Zhou, Yangning Li, Zhongli Li, Ruiyang Liu, Rongyi Sun, Zizhen Wang, Chao Li, Yunbo Cao, and Hai-Tao Zheng. 2022. The past mistake is the future wisdom: Error-driven contrastive probability optimization for chinese spell checking. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3202–3213.

Zhenqing Ling, Yuexiang Xie, Chenhe Dong, and Ying Shen. 2025. Enhancing factual consistency in text summarization via counterfactual debiasing. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 7912–7924.

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

Da Ma, Lu Chen, Situo Zhang, Yuxun Miao, Su Zhu, Zhi Chen, Hongshen Xu, Hanqi Li, Shuai Fan, Lei Pan, and 1 others. 2024. Compressing kv cache for long-context llm inference with inter-layer attention similarity. *arXiv preprint arXiv:2412.02252*.

Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, and 1 others. 2023. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Betty Van Aken, Benjamin Winter, Alexander Löser, and Felix A Gers. 2019. How does bert answer questions? a layer-wise analysis of transformer representations. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1823–1832.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

Qiuchen Wang, Ruixue Ding, Zehui Chen, Weiqi Wu, Shihang Wang, Pengjun Xie, and Feng Zhao. 2025. Vidorag: Visual document retrieval-augmented generation via dynamic iterative reasoning agents. *arXiv preprint arXiv:2502.18017*.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. Efficient streaming language models with attention sinks. In *The Twelfth International Conference on Learning Representations*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Wang, Bowen Yu, Chengpeng Li, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380.

Fanghua Ye, Jarana Manotumruksa, and Emine Yilmaz. 2022. Multiwoz 2.4: A multi-domain task-oriented dialogue dataset with essential annotation corrections to improve state tracking evaluation. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 351–360.

Zihao Yi, Jiarui Ouyang, Yuwen Liu, Tianhao Liao, Zhe Xu, and Ying Shen. 2024. A survey on recent advances in llm-based multi-turn dialogue systems. *arXiv preprint arXiv:2402.18013*.

Zihao Yi, Zhe Xu, and Ying Shen. 2025. Intent-driven in-context learning for few-shot dialogue state tracking. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Jiahao Zhang, Haiyang Zhang, Dongmei Zhang, Liu Yong, and Shen Huang. 2024. End-to-end beam retrieval for multi-hop question answering. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1718–1731.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, and 1 others. 2025a. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

Zhiqiang Zhang, Liqiang Wen, and Wen Zhao. 2025b. Rule-kbqa: Rule-guided reasoning for complex knowledge base question answering with large language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 8399–8417.

## Technical Appendix

The appendix is organized as follows:

- **Sec. A: Implementation of experiments**.
- **Sec. B: Multi-document attention distributions**. Attention distributions across models with varying numbers of input documents.
- **Sec. C: Theoretical analysis of AttnRank**, comprising:
  - Sec. C.1 Optimization objective and problem statement
  - Sec. C.2 Key assumptions
  - Sec. C.3 Technical lemmas on hidden-state decomposition and logit formation
  - Sec. C.4 Main proposition
  - Sec. C.5 Corollary on optimal document positioning and empirically verified hypothesis on layer-wise effects
  - Sec. C.6 Connection between our theory and **AttnRank**
- **Sec. D: Attention score computation**. Formal definition of the attention score used for document-level profiling.
- **Sec. E: Data volume requirements**. Convergence analysis of data requirements for reliably characterizing LLM attention distribution patterns.
- **Sec. F: Supplementary experiments and case studies.**
- **Sec. H: Generalization to latest models.** Results on the Qwen3 model family demonstrating **AttnRank**'s effectiveness on the latest LLMs.

## A Implementation of experiments

**Advantages of Our Method** The **AttnRank** framework is designed to be both effective and practical, offering several key advantages:

- **Training-Free and Model-Agnostic: AttnRank** requires no modification to the LLM's architecture or parameters. It treats the model as a black box, making it universally applicable to any Transformer-based LLM.
- **Extremely Low Overhead:** The primary cost is a **one-time profiling step**, which, as we've shown, requires a minimal number of inference runs. Once the attention profile is saved, it can be reused for all future inference tasks for that model at virtually no cost. The reranking itself is a simple array permutation, which is computationally negligible.
- **Compatibility with Modern Acceleration Frameworks:** Because **AttnRank** is an input-preprocessing step that operates before the main generation pass, it is fully compatible with popular inference acceleration libraries like **Flash Attention** and serving frameworks like **vLLM**. It does not interfere with their internal optimizations, allowing users to gain performance from our method while retaining the benefits of these high-speed tools.

**Experimental setup** All experiments were conducted on the same hardware configuration, as detailed in Table 4. In the first stage of **AttnRank**, we use the `AutoModelForCausalLM` function from the HuggingFace Transformers library to load the model, perform inference, and extract attention distributions. In the second stage, we adopt the VLLM framework (Kwon et al., 2023) to accelerate inference.

| Architecture | x86_64 |
|---|---|
| **CPU** | Intel Xeon Gold 5218R @ 2.10GHz |
| **GPU** | NVIDIA GeForce RTX 3090 24GB × 10 |
| **CUDA Toolkit** | 11.3 |
| **Operating System** | Ubuntu 20.04 |
| **Programming Language** | Python 3.9.18 |
| **Deep Learning Framework** | PyTorch 1.13.0 |

Table 4: Experimental Environment Configuration

### Utilized LLMs

- **DeepSeek-R1-Distill-Llama-8B** (Guo et al., 2025): An 8B-parameter model distilled from DeepSeek-R1, fine-tuned via reinforcement learning to enhance reasoning capabilities.
- **DeepSeek-LLM** (Bi et al., 2024): A 7B parameter models trained on 2 trillion tokens, utilizing a pre-norm decoder-only Transformer architecture with grouped-query attention (GQA).
- **LLaMA-3 Series** (Grattafiori et al., 2024): Meta's LLaMA 3 series includes 8B and 70B parameter models trained on 15 trillion tokens.The instruct variant is fine-tuned to enhance instruction-following capabilities.

- **LLaMA-2-13B-hf** (Touvron et al., 2023): A 13B parameter decoder-only Transformer model, trained on 2 trillion tokens, featuring a 4,096-token context window optimized for general-purpose language tasks.
- **Mistral-7B-Instruct** (Jiang et al., 2023): A 7B parameter model fine-tuned for instruction following, employing grouped-query attention and sliding window attention mechanisms to handle long sequences.
- **InternLM3-8B-Instruct** (Cai et al., 2024): An 8B parameter model fine-tuned to follow instructions, designed to perform effectively across a variety of natural language understanding tasks.
- **Qwen 2.5 Series** (Yang et al., 2024): Developed by Alibaba, the Qwen 2.5 series includes models with 1.5B, 3B, and 7B parameters, trained on extensive datasets to support multilingual capabilities and demonstrate strong performance across diverse tasks.
- **Code LLaMA 7B** (Roziere et al., 2023): A code-specialized model fine-tuned from LLaMA 2, supporting code generation tasks across multiple programming languages.
- **Qwen3 Series** (Yang et al., 2025): The latest generation of the Qwen model family, including Qwen3-0.6B, Qwen3-4B, Qwen3-4B-Instruct, Qwen3-4B-Math, Qwen3-4B-EFT, and the mixture-of-experts variant Qwen3-30B-A3B. These models span a range of sizes and specializations, enabling evaluation of **AttnRank** across diverse training regimes.

**Utilized Datasets**

- **HotpotQA** (Yang et al., 2018) is a large-scale dataset designed to facilitate research in multi-hop question answering. It comprises approximately 113,000 question-answer pairs, each requiring reasoning over multiple Wikipedia articles to derive the correct answer.
- **2WikiMultiHopQA** (Ho et al., 2020) is a dataset constructed to assess the comprehensive reasoning abilities of question answering models. It contains question-answer pairs that require multi-hop reasoning over both structured data from Wikidata and unstructured text from Wikipedia.
- **MultiWOZ** (Eric et al., 2020; Ye et al., 2022) series are multi-domain task-oriented dialogue datasets with annotated dialogue states. Each dialogue covers 1 to 5 domains, with an average of 13.7 turns per dialogue.

**Prompts** As shown in Figure 5 and Figure 6, we present the prompt templates used in our experiments. Retrieved documents and contextual examples are inserted into `Document [1]` and `Example [1]` positions after reordering by different strategies to compare their impact on performance.

**Input Context**

Write a high-quality answer for the given question using only the provided search results (some of which might be irrelevant).

{Example}

Document [1] The first Nobel Prize in Physics was to Wilhelm Conrad Röntgen …
.....
Document [5] The Nobel Prize in Physics is a yearly award given by …

Question: who got the first nobel prize in physics
Answer:

Figure 5: Prompt for multi-hop QA experiment.

**Input Context**

CREATE TABLE attraction(
name text, area text, type text)
… …

Example [1]:
Context: hotel-stars: 4, hotel-area: west
… …
SQL: SELECT * FROM attraction WHERE area = west
Example [2]:
… …

Context: restaurant-book time: 15:30, restaurant-area: centre
System: Booking was successful. Anything else?
User: Could you also find me some places to go in the same area as the restaurant?
SQL:

Figure 6: Prompt for dew-shot experiment.

As shown in Figure 7, we provide the prompt template used in the analysis with disrupted input structure in Section 3 . Twenty documents are randomly shuffled and inserted, with all punctuation removed and uppercase letters converted to lowercase to destroy structural cues.

**Input Context**

write a high-quality answer for the given question using only the provided search results some of which might be irrelevant

{example}

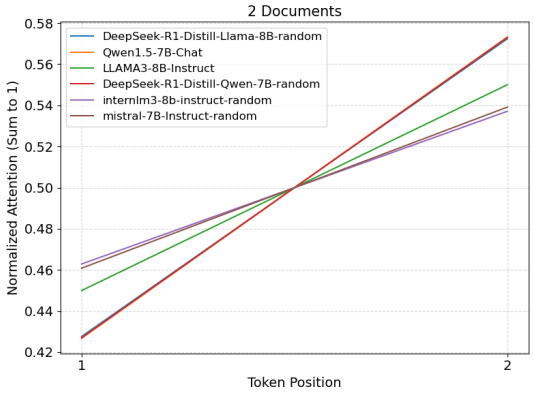
the first nobel prize in physics was to wilhelm conrad röntgen …
.....
the nobel prize in physics is a yearly award given by …

who got the first nobel prize in physics
answer

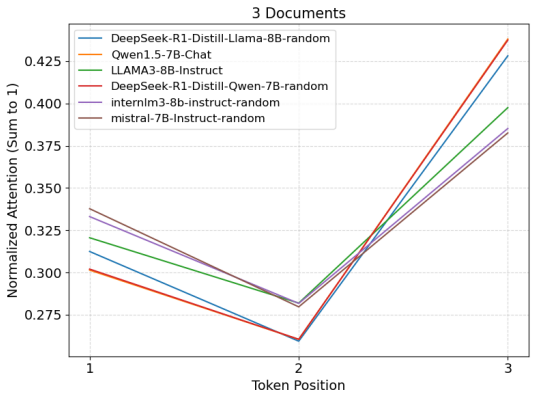
Figure 7: Prompt after disrupting the structural blocks

## B More details of how LLMs distribute their attention.

Figures 8, 9, and 10 show the attention distributions across different models and document counts, illustrating that the attention basin phenomenon is consistently observed.




(a) Two documents

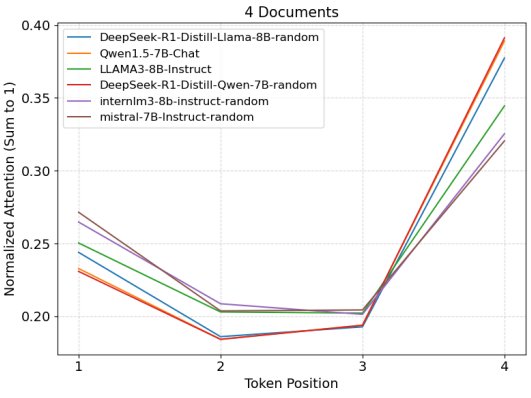



(b) Three documents

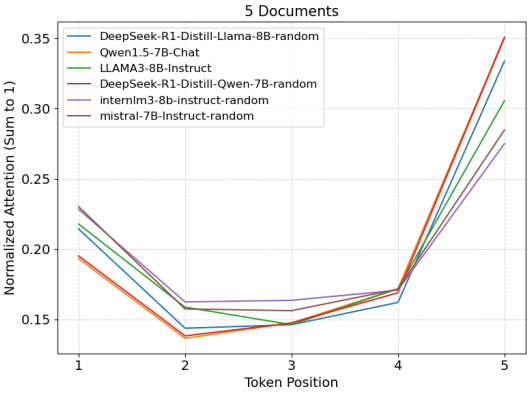
Figure 8: Attention distribution with 2 and 3 documents

## C Theoretical analysis of attention-guided reranking in long-context tasks

The following analysis is intended to provide *mechanistic intuition* for why attention-guided reranking can improve long-context performance. To keep the derivation tractable, we adopt several idealized assumptions that may not strictly hold in real LLMs. In practice, (i) document representations are generally *correlated* rather than orthogonal, (ii) attention is not purely position-separable and exhibits strong *content–position interactions*, and (iii) the output logit formation does not exactly decompose into clean "document" and "token" subspaces. Nevertheless, these assumptions capture a first-order effect: *increasing the effective attention mass on the answer-bearing document tends to increase its contribution to the final hidden state and thus the probability of the correct answer*. Below we explicitly identify the idealizations used and explain how the conclusions remain qualitatively meaningful under weaker, more realistic conditions.




(a) Four documents



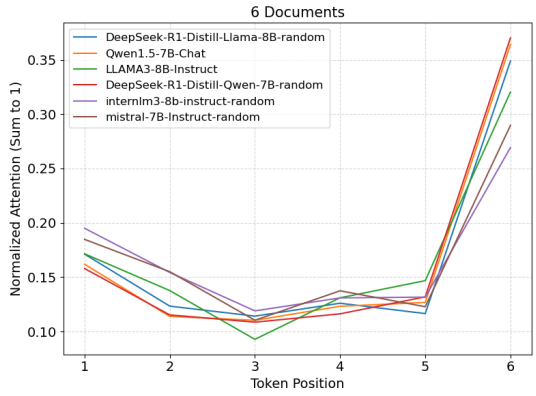

(b) Five documents

Figure 9: Attention distribution with 4 and 5 documents

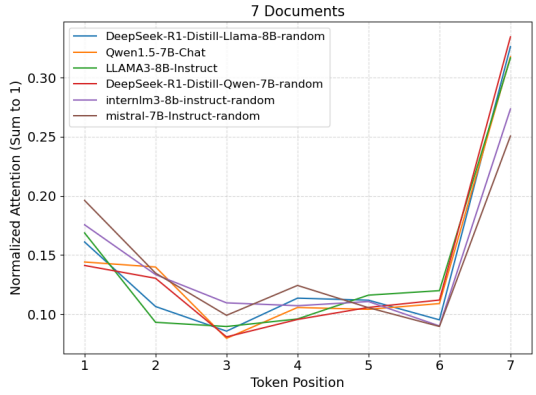
### C.1 Optimization objective

Given a long-context input $S = \{t, d_1, \ldots, d_n, q\}$, where $t$ is a fixed prompt template, $q$ is the user query, and $D = \{d_1, \ldots, d_n\}$ denotes information segments containing large numbers of tokens, such as knowledge documents in multi-hop QA or in-context examples in few-shot tasks. Let $d^*$ denote the correct document, we aim to demonstrate that strategically positioning $d^*$ in high-attention regions of the input sequence increases its cross-layer average attention weight $\bar{\alpha}_{d^*}$, thereby maximizing the posterior probability $P(y^*|x, D)$ of generating correct answer $y^*$. Formally, we seek to prove:

$$\bar{\alpha}_{d^*} > \bar{\alpha}_{d_j} \quad (\forall d_j \neq d^*)$$
$$\Longrightarrow P(y^* \mid x, D) > P(y \mid d_j, x, D) \qquad (3)$$




(a) Six documents




(b) Seven documents

Figure 10: Attention distribution with 6 and 7 documents

### C.2 Key assumptions

**Assumption C.1** (Orthogonal Semantic Representation)**.** Let $\{e_{d_k}\}$ denote document-level embeddings. These satisfy pairwise orthogonality:

$$\langle e_{d_i}, e_{d_j} \rangle = 0 \quad \forall i \neq j \tag{4}$$

with $\|e_{d_k}\|_2 = 1$ for normalization.

**Assumption C.2** (Position-Attention Coupling)**.** Let $\alpha_p^{(l)}$ denote attention weight for position $p$ at layer $l$. There exists position-dependent bias:

$$\mathbb{E}[\alpha_p^{(l)}] = f(p) + \epsilon_p^{(l)} \tag{5}$$

where $f : \mathbb{N} \to \mathbb{R}^+$ is a U-shaped function modeling the attention basin phenomenon, and $\epsilon_p^{(l)}$ represents position-agnostic content-based attention.

**Assumption C.3** (Compositional Weight Binding)**.** The output projection matrix $W_y$ decomposes as:

$$W_y = [E_d \| E_t] \cdot W_c \tag{6}$$

where $E_d$ is the document embedding matrix, $E_t$ the token embedding matrix, and $W_c$ a composition matrix satisfying $\|W_c\|_F \leq \gamma$.

### C.3 Technical lemmas

**Lemma C.4** (Hidden State Composition)**.** *The final hidden state $h_{last}$ decomposes into document-aware components:*

$$h_{last} = h_{init} + \sum_{k=1}^{n} \left( \sum_{l=1}^{L} \alpha_{d_k}^{(l)} v_{d_k}^{(l)} \right) + \Delta h_{noise} \tag{7}$$

*where $\alpha_{d_k}^{(l)} = \sum_{p \in \mathcal{P}(d_k)} \alpha_p^{(l)}$ aggregates position-wise attention for document $d_k$, and $v_{d_k}^{(l)}$ represents its value vector at layer $l$.*

*Proof.* Through residual connections, each layer's output accumulates document-specific contributions:

$$\begin{aligned} h^{(l)} &= h^{(l-1)} + \text{Attn}^{(l)}(h^{(l-1)}) \\ &= h^{(0)} + \sum_{m=1}^{l} \text{Attn}^{(m)}(h^{(m-1)}) \end{aligned} \tag{8}$$

Decompose attention heads into document-level components:

$$\text{Attn}^{(m)} = \sum_{k=1}^{n} \alpha_{d_k}^{(m)} v_{d_k}^{(m)} + \text{CrossDoc}^{(m)} \tag{9}$$

Under Assumption C.1, cross-document terms $\text{CrossDoc}^{(m)}$ become negligible due to orthogonality, yielding the stated decomposition. □

**Lemma C.5** (Logit Formation)**.** *The logit for answer token $y^*$ decomposes as:*

$$logit(y^*) = \underbrace{\langle h_{last}, e_{d^*} \rangle}_{\textit{document term}} + \underbrace{\langle h_{last}, e_{y^*} \rangle}_{\textit{token term}} + b_{y^*} \tag{10}$$

*where $e_{d^*}$ and $e_{y^*}$ are orthogonal components from Assumption C.3.*

*Proof.* Using Assumption C.3, the output projection becomes:

$$\begin{aligned} W_y h_{\text{last}} &= [E_d \| E_t] W_c h_{\text{last}} \\ &= E_d(W_c^{(d)} h_{\text{last}}) + E_t(W_c^{(t)} h_{\text{last}}) \end{aligned} \tag{11}$$

Thus for target token $y^*$ embedded as $e_{y^*} = E_t[i]$, the logit contains separate document and token alignment terms. □

### C.4 Main proposition

**Proposition C.6** (Attention-Probability Monotonicity)**.** *For documents $d^*$ and $d_j$ with average attention weights $\bar{\alpha}_{d^*} = \frac{1}{L}\sum_{l=1}^{L} \alpha_{d^*}^{(l)}$ and $\bar{\alpha}_{d_j}$, if $\bar{\alpha}_{d^*} > \bar{\alpha}_{d_j}$, then:*

$$\frac{\partial P(y^*|x, D)}{\partial \bar{\alpha}_{d^*}} > \frac{\partial P(y^*|x, D)}{\partial \bar{\alpha}_{d_j}} \geq 0 \tag{12}$$

*Proof.* **Step 1: Document-term dominance** From Lemma C.4 and C.5, the document alignment term dominates when $d^*$ contains sufficient answer evidence:

$$\langle h_{\text{last}}, e_{d^*} \rangle = \bar{\alpha}_{d^*} \underbrace{\left\langle \frac{1}{L}\sum_{l=1}^{L} v_{d^*}^{(l)}, e_{d^*} \right\rangle}_{\kappa} + \mathcal{O}(\max_{k \neq *} \bar{\alpha}_{d_k}) \tag{13}$$

where $\kappa > 0$ due to value-key alignment in transformers.

**Step 2: Probability gradient analysis** The output probability computes as:

$$P(y^*|x, D) = \frac{\exp(\text{logit}(y^*))}{\sum_y \exp(\text{logit}(y))} \tag{14}$$

Taking partial derivative with respect to $\bar{\alpha}_{d^*}$:

$$\frac{\partial P}{\partial \bar{\alpha}_{d^*}} = P(y^*|x, D)(1 - P(y^*|x, D))\kappa > 0 \tag{15}$$

Similarly, $\frac{\partial P}{\partial \bar{\alpha}_{d_j}} = -P(1-P)\kappa_j \leq 0$ for $j \neq *$.

**Step 3: Strict monotonicity** Given $\kappa \propto \|v_{d^*}\| \cos\theta_{v_{d^*}, e_{d^*}}$ and Assumption C.1, $\cos\theta = 1$. Thus:

$$\bar{\alpha}_{d^*} > \bar{\alpha}_{d_j} \implies \frac{\partial P}{\partial \bar{\alpha}_{d^*}} > \left| \frac{\partial P}{\partial \bar{\alpha}_{d_j}} \right| \tag{16}$$

Hence complete the proof. □

### C.5 Implications for long-context tasks

**Corollary C.7** (Optimal Document Positioning)**.** *Let $\mathcal{P}_{opt}$ denote positions with maximal attention basin effect in Assumption C.2. Placing $d^*$ at $p^* \in \mathcal{P}_{opt}$ maximizes $\bar{\alpha}_{d^*}$, leading to:*

$$\mathbb{E}[P(y^*|x, D)]_{p^*} \geq \mathbb{E}[P(y^*|x, D)]_p \quad \forall p \notin \mathcal{P}_{opt} \tag{17}$$

*Proof.* From Assumption C.2, positions in $\mathcal{P}_{\text{opt}}$ maximize $\mathbb{E}[\alpha_p^{(l)}]$. By the monotonicity in Proposition 1, positioning $d^*$ at $p^*$ achieves:

$$\begin{aligned} \mathbb{E}[\bar{\alpha}_{d^*}|p^*] &= \frac{1}{L}\sum_{l=1}^{L} f(p^*) + \mathbb{E}[\epsilon_{p^*}^{(l)}] \\ &> \frac{1}{L}\sum_{l=1}^{L} f(p) + \mathbb{E}[\epsilon_p^{(l)}] \end{aligned} \tag{18}$$

for any $p \notin \mathcal{P}_{\text{opt}}$. Proposition 1 then guarantees higher $P(y^*|x, D)$. □

*Hypothesis* C.8 (Layer-wise Attention Degradation). Let $\rho(l) = \frac{\mathbb{V}[\epsilon_p^{(l)}]}{\mathbb{V}[f(p)]}$ measure the relative strength of content-based vs position-based attention at layer $l$. There exists a layer depth threshold $L^*$ such that:

$$\begin{aligned} &\rho(l) < 1 \quad \forall l < L^* \quad \textit{(position-dominated regime)} \\ &\rho(l) \geq 1 \quad \forall l \geq L^* \quad \textit{(content-dominated regime)} \end{aligned}$$

Thus, shallow layers better preserve positional bias patterns from Assumption C.2, while deeper layers exhibit attenuated positional effects.

**Empirical motivation and verification.** From Assumption C.2, the attention variance at layer $l$ can be decomposed as:

$$\mathbb{V}[\alpha_p^{(l)}] = \underbrace{\mathbb{V}[f(p)]}_{\text{positional}} + \underbrace{\mathbb{V}[\epsilon_p^{(l)}]}_{\text{content-based}} \tag{19}$$

Empirically, Transformer architectures exhibit increasing $\mathbb{V}[\epsilon_p^{(l)}]$ with depth as self-attention becomes more semantic (Abnar and Zuidema, 2020). When $\mathbb{V}[\epsilon_p^{(l)}]$ surpasses $\mathbb{V}[f(p)]$ at some layer $L^*$, positional effects become subdominant. We verify this hypothesis experimentally in Section 3 (Figure 3), where reranking based on shallow-layer attention consistently outperforms deeper layers, confirming the predicted transition from position-dominated to content-dominated regimes.

### C.6 Remark: connecting theory to method

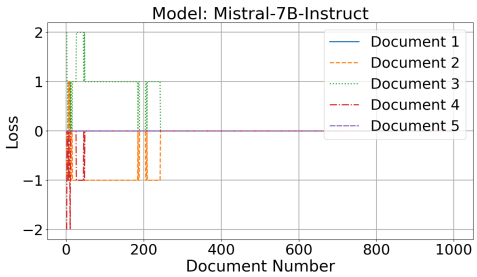
Our theoretical analysis rigorously justifies the core principle of attention-guided document reranking:

- The U-shaped attention basin (Assumption C.2) explains the *lost-in-the-middle* phenomenon through its positional expectation $\mathbb{E}[\alpha_p^{(l)}]$
- The attention-probability monotonicity (Proposition 1) formally establishes that boosting $\bar{\alpha}_{d^*}$ via strategic positioning directly increases answer correctness

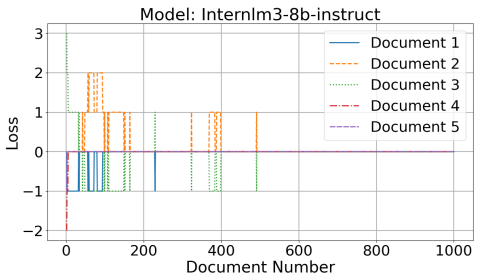
- Corollary C.7 provides theoretical guarantees for our method's effectiveness: reranking documents to place $d^*$ in $\mathcal{P}_{\text{opt}}$ (typically sequence edges) maximizes its attention influence

- Hypothesis C.8 captures the layer-dependent nature of attention guidance: shallow layers' position-dominated regime ($\rho(l) < 1$) better preserves document ordering signals critical for reranking, while deeper layers' content-focused attention ($\rho(l) \geq 1$) introduces positional ambiguity. This empirically verified hypothesis explains why using shallower attention maps produces better reranking results.

This mathematical foundation not only explains empirical observations but also guides future extensions: the composition matrix $W_c$ in Assumption C.3 suggests directions for training-based attention shaping, while the orthogonality in Assumption C.1 motivates improved document encoding schemes to better satisfy theoretical prerequisites.




(a) Mistral-7B-Instruct




(b) Internlm3-8b-instruct

Figure 11: The relationship between long-context data volume and attention distribution in LLMs.

## D Attention Score Computation

In this section, we formalize the attention score computation used to profile each document's attention weight.

Let $Q$ denote the set of question/answer token indices, $D_k$ denote the set of token indices for document $k$, and $A_{i,j}$ denote the attention weight from token $i$ to token $j$. The attention score for document $k$ is computed as:

$$\text{Score}(D_k) = \frac{1}{|Q|} \sum_{i \in Q} \sum_{j \in D_k} A_{i,j} \tag{20}$$

This formulation measures the average attention that all query/answer tokens collectively pay to the tokens within document $k$. A higher score indicates that the model allocates more attention to that document, suggesting its content is more salient to the model during generation. This score is the basis for the attention profiling step described in Section 4.

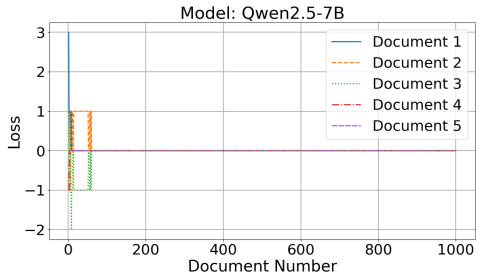
## E How many documents are required?

We design an experiment to determine how much data is required for attention patterns to converge. Following the setup in Section 5.1, we incrementally increase the number of samples and compare the resulting attention distributions to those from the full dataset. As shown in Figures 11 and 12, all models converge after about 400 samples. Notably, we focus on convergence at the document boundaries—the first (blue) and last (purple) documents. In DeepSeek-LLM and InternLM3-8B-Instruct, boundary attention converges with only 200 samples. Mistral-7B-Instruct and Qwen2.5-7B match the final pattern from the start. These results suggest that, for some models, a single sample may suffice to approximate full-context attention behavior.

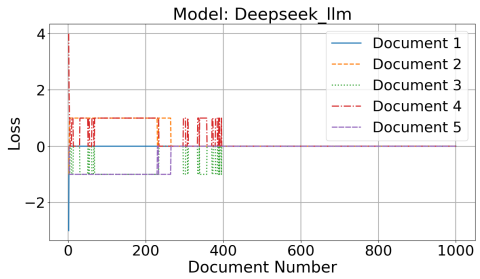
## F Attention distribution experiments and case studies

Following the setup in Experiment 5.1, we analyze the average attention scores assigned to ground-truth and noise documents under different reranking baselines. As shown in Table 7, **AttnRank** assigns the highest average attention score to relevant documents and the lowest to noise ones, demonstrating that **AttnRank** effectively guides the model to focus on the most critical documents.

We further conducted three case studies. Figures 14–13, 15–16 and 17–18 present two additional case studies. **AttnRank** robustly maintains high attention on critical documents and low attention on noise, validating its effectiveness. In contrast, the descending strategy preserves high attention for relevant documents but also highlights noise, and the lost-in-the-middle approach reduces noise




(a) Qwen2.5-7B




(b) Deepseek-llm

Figure 12: The relationship between sample size and attention distribution in LLMs.

| Models | Descending | Ascending | AttnRank (Ours) |
|---|---|---|---|
| llama2-13b-hf | 0.264 | **0.295** | 0.285 |
| deepseek-llm | 0.255 | **0.311** | 0.269 |
| R1-Distill-Llama-8B | 0.162 | **0.167** | 0.160 |
| internlm3-8b-instruct | **0.231** | 0.217 | 0.221 |
| llama3-8b-instruct | 0.223 | 0.246 | **0.268** |
| llama3-8b | 0.312 | **0.331** | 0.326 |
| mistral-7B-Instruct | 0.287 | **0.300** | 0.283 |
| Qwen2.5-1.5B-Instruct | 0.221 | **0.259** | 0.255 |
| Qwen2.5-3B-Instruct | 0.323 | **0.347** | 0.346 |
| Qwen2.5-7B-Instruct | 0.366 | **0.394** | 0.390 |
| **Average** | 0.264 | **0.287** | 0.280 |

Table 5: Performance on **2WikiMultiHopQA** with destructured inputs.

| Models | Descending | Ascending | AttnRank (Ours) |
|---|---|---|---|
| llama2-13b-hf | 0.346 | **0.369** | 0.347 |
| deepseek-llm | 0.379 | **0.409** | 0.371 |
| R1-Distill-Llama-8B | 0.179 | **0.188** | 0.179 |
| internlm3-8b-instruct | **0.313** | 0.311 | 0.307 |
| llama3-8b-instruct | 0.264 | **0.311** | 0.275 |
| llama3-8b | 0.407 | 0.427 | **0.433** |
| mistral-7B-Instruct | 0.366 | **0.398** | 0.379 |
| Qwen2.5-1.5B-Instruct | 0.354 | **0.366** | 0.337 |
| Qwen2.5-3B-Instruct | 0.450 | **0.467** | 0.455 |
| Qwen2.5-7B-Instruct | 0.470 | **0.491** | 0.482 |
| **Average** | 0.353 | **0.374** | 0.357 |

Table 6: Performance on **HotpotQA** with destructured inputs.

attention at the expense of under-attending to relevant documents. **AttnRank** consistently focuses on important content while disregarding noise.

# G Model Behavior under Destructured Context Blocks

We further analyze model behavior after *destroying block-level structure*, including removing document delimiters, punctuation, and capitalization, and concatenating all documents into a single plain text sequence. This setting eliminates explicit structural cues that give rise to the U-shaped attention basin discussed in the main paper.

## G.1 Results on Destructured Inputs

## G.2 Key Observations

After removing structural cues, the attention basin disappears and models exhibit a strong recency bias. Consequently, the **Ascending** strategy (placing the most relevant document last) achieves the best average performance on both datasets.

Although AttnRank is optimized for the U-shaped attention basin, it remains competitive even after structure removal, but is consistently outperformed by Ascending under this setting.

Notably, the performance gap between Descending and Ascending increases after structure destruction:

- **HotpotQA:** Improvement increases from 0.92% (with structure) to 2.09% (without structure)
- **2WikiMultiHopQA:** Improvement increases from 1.65% (with structure) to 2.23% (without structure)

These results further reinforce our core claim: *the attention basin is structure-dependent*, and reranking strategies must be aligned with the model's effective attention profile, which changes substantially when structural cues are removed.

# H Generalization to Latest Models

To address concerns about generalization to more recent model families, we repeated the multi-hop QA experiment (HotpotQA setup from Section 5.1) on the latest Qwen3 series (Yang et al., 2025), which includes base, instruct, math-specialized, and mixture-of-experts variants. The results are presented in Table 8.

**Key Observations.**

- **AttnRank achieves the highest average accuracy (0.502)** across all tested Qwen3 models, outperforming Ascending (0.494), Descending (0.488), and LIM (0.484).

Table 7: Average attention scores on ground-truth and noise documents under different reranking strategies.

| Model | Correct Documents ↑ | | | | | Wrong Documents ↓ | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Random | Descending | Ascending | LIM | Rerank | Random | Descending | Ascending | LIM | **AttnRank (ours)** |
| DeepSeek-R1-Distill-Llama-8B | 0.0182 | 0.0183 | **0.0194** | 0.0183 | 0.0193 | 0.0048 | 0.0048 | 0.0043 | 0.0048 | **0.0042** |
| LLAMA3-8B-Instruct | 0.0212 | 0.0212 | 0.0230 | 0.0214 | **0.0232** | 0.0054 | 0.0054 | 0.0045 | 0.0053 | **0.0043** |
| LLAMA3-8B | 0.0238 | 0.0238 | **0.0258** | 0.0238 | 0.0256 | 0.0062 | 0.0062 | 0.0053 | 0.0062 | **0.0052** |
| mistral-7B-Instruct | 0.0230 | 0.0229 | 0.0238 | 0.0231 | **0.0244** | 0.0059 | 0.0060 | 0.0056 | 0.0059 | **0.0051** |
| deepseek-llm | 0.0389 | 0.0387 | 0.0401 | 0.0386 | **0.0404** | 0.0110 | 0.0111 | 0.0106 | 0.0112 | **0.0103** |
| internlm3-8b-instruct | 0.0177 | 0.0178 | 0.0181 | 0.0177 | **0.0185** | 0.0045 | 0.0046 | 0.0044 | 0.0045 | **0.0041** |
| qwen 2.5 1.5B | 0.0118 | 0.0119 | **0.0128** | 0.0118 | 0.0127 | 0.0030 | 0.0031 | **0.0026** | 0.0030 | **0.0026** |
| qwen 2.5 3B | 0.0169 | 0.0168 | 0.0170 | 0.0166 | **0.0183** | 0.0043 | 0.0043 | 0.0044 | 0.0044 | **0.0034** |
| qwen 2.5 7B | 0.0283 | 0.0285 | **0.0305** | 0.0284 | 0.0304 | 0.0072 | 0.0072 | **0.0061** | 0.0072 | **0.0061** |
| **Average** | 0.022200 | 0.022211 | 0.023389 | 0.022189 | **0.023644** | 0.005811 | 0.005856 | 0.005311 | 0.005833 | **0.005033** |

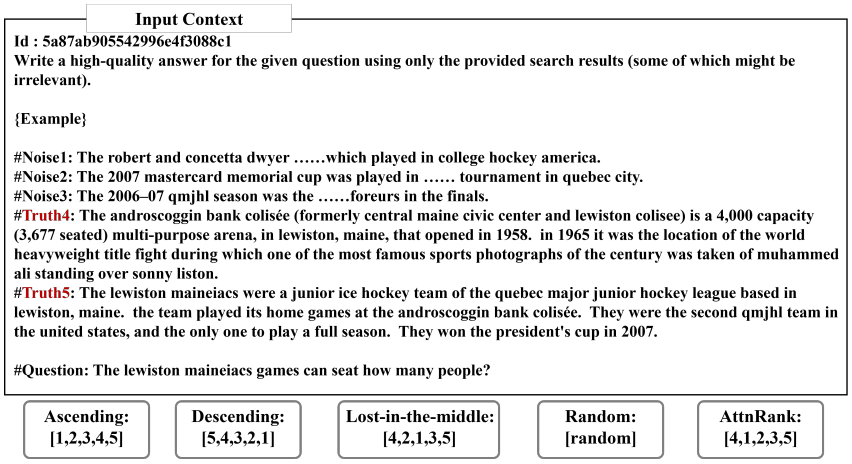



Figure 13: Case 1's input prompt and ranking outcomes under different reranking strategies.

| **Model** | **Ascending** | **Descending** | **LIM** | **AttnRank (Ours)** |
|---|---|---|---|---|
| Qwen3-0.6B | 0.320 | 0.281 | 0.277 | **0.325** |
| Qwen3-4B | 0.541 | 0.553 | 0.554 | **0.557** |
| Qwen3-4B-Instruct | **0.544** | **0.544** | 0.527 | 0.542 |
| Qwen3-4B-Math | 0.529 | 0.528 | 0.533 | **0.547** |
| Qwen3-4B-EFT | 0.532 | 0.543 | 0.528 | 0.542 |
| Qwen3-30B-A3B | **0.500** | 0.478 | 0.486 | **0.500** |
| **Average** | 0.494 | 0.488 | 0.484 | **0.502** |

Table 8: Answer accuracy on HotpotQA using the Qwen3 model family under different document ordering strategies.

- **Improvement is consistent: AttnRank** yields the best or tied-best performance on 4 out of 6 models, demonstrating robust effectiveness across diverse model variants.

- **Domain fine-tuning does not erase the effect:** Models fine-tuned for specific tasks such as mathematics (Qwen3-4B-Math) still benefit from attention-guided reranking, confirming that the attention basin phenomenon persists across training regimes.

These results confirm that **AttnRank** generalizes effectively to the latest generation of LLMs, including mixture-of-experts architectures (Qwen3-30B-A3B), further validating the universality of the attention basin phenomenon and the practical value of our method.

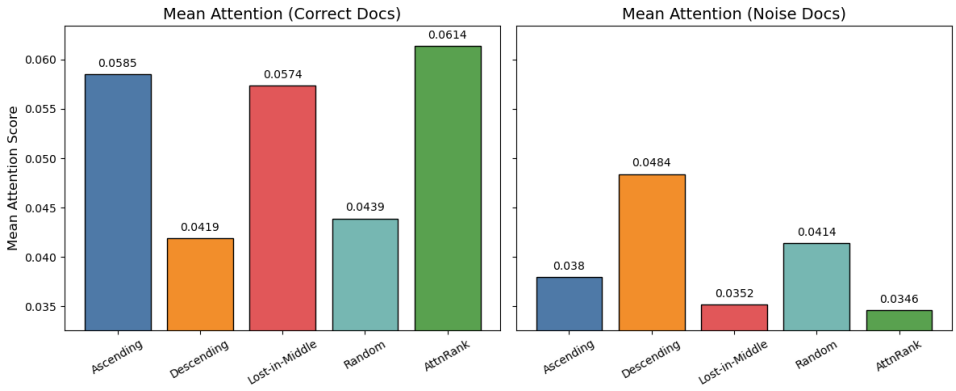



Figure 14: Average attention scores for relevant and noise documents under various reranking strategies in case 1. **AttnRank** attains the highest attention on relevant documents and the lowest on irrelevant documents, validating its effectiveness.

**Input Context**

**Id : 5a8e3ea95542995a26add48d**
**Write a high-quality answer for the given question using only the provided search results (some of which might be irrelevant).**

**{Example}**

**#Noise1: Hamish & andy's gap year is a…… logie called 'the raintree'.**
**#Noise2: Kingston morning is dave eggar's ……best instrumental arrangement".**
**#Noise3: Nola is a 2003 american ……new york city on july 23, 2004.**
**#Truth4: Big stone gap is a 2014 american drama romantic comedy film written and directed by adriana trigiani and produced by donna gigliotti for altar identity studios, a subsidiary of media society. Based on trigiani's 2000 best-selling novel of the same name, the story is set in the actual virginia town of big stone gap circa 1970s. The film had its world premiere at the virginia film festival on november 6, 2014.**
**#Truth5: Adriana trigiani is an italian american best-selling author of sixteen books, television writer, film director, and entrepreneur based in greenwich village, new york city. Trigiani has published a novel a year since 2000.**

**#Question: The director of the romantic comedy "big stone gap" is based in what new york city?**

| **Ascending:** [1,2,3,4,5] | **Descending:** [5,4,3,2,1] | **Lost-in-the-middle:** [4,2,1,3,5] | **Random:** [random] | **AttnRank:** [4,1,2,3,5] |
|---|---|---|---|---|

Figure 15: Case 2's input prompt and ranking outcomes under different reranking strategies.

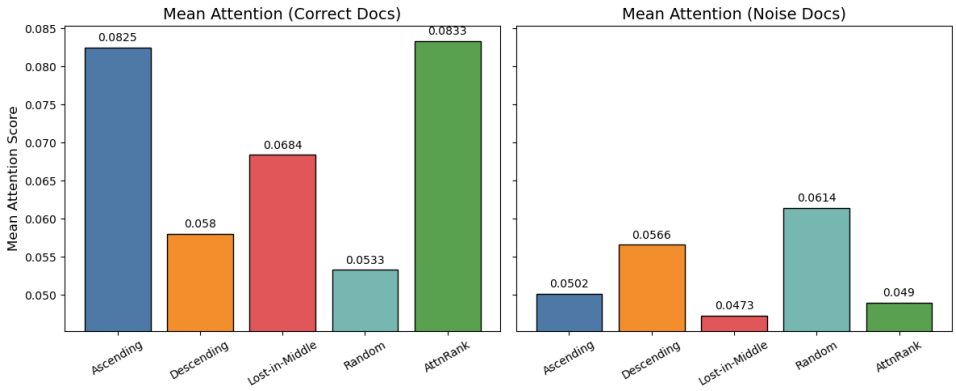



Figure 16: Average attention scores for relevant and noise documents in case 2.

**Input Context**

**Id : 5a87ab905542996e4f3088c1**
**Write a high-quality answer for the given question using only the provided search results (some of which might be irrelevant).**

**{Example}**

**#Noise1: Document1: Esma Sultan (14 March 1726 – 13 August 1788) ……III and Abdul Hamid I.**
**#Noise2: Document2: Esma Sultan (17 July 1778 – 4 June 1848) ……Sultan and Rahime Perestu Sultan.**
**#Noise3: Document3: Esma Sultan is the name of three daughters of three Ottoman Sultans:**
**#Truth4: Document4: The Esma Sultan Mansion (Turkish: "Esma Sultan Yalısı" ), a historical yalı (English: waterside mansion ) located at Bosphorus in Ortaköy neighborhood of Istanbul, Turkey and named after its original owner Esma Sultan, is used today as a cultural center after being redeveloped.**
**#Truth5: Document5: The Laleli Mosque (Turkish: "Laleli Camii, or Tulip Mosque" ) is an 18th-century Ottoman imperial mosque located in Laleli, Fatih, Istanbul, Turkey.**
**#Question: Question Are the Laleli Mosque and Esma Sultan Mansion located in the same neighborhood?**

| Ascending: [1,2,3,4,5] | Descending: [5,4,3,2,1] | Lost-in-the-middle: [4,2,1,3,5] | Random: [random] | AttnRank: [4,1,2,3,5] |
|---|---|---|---|---|

Figure 17: Case 3's input prompt and ranking outcomes under different reranking strategies.

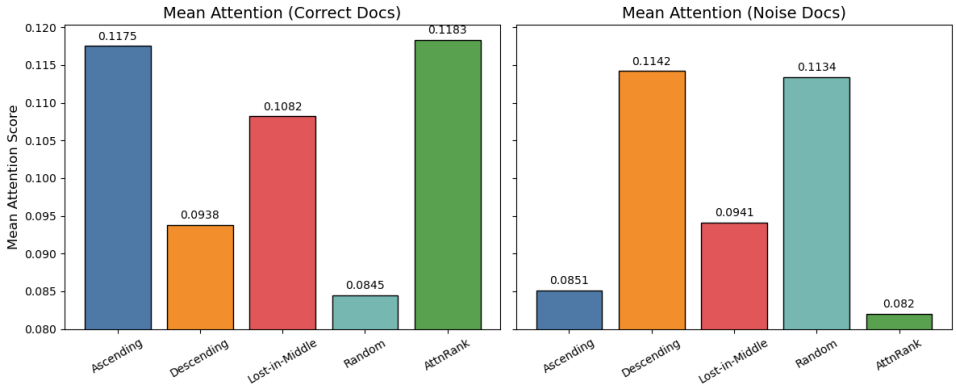



Figure 18: Average attention scores for relevant and noise documents in case 3.