POSITION BIAS CORRECTION IS INSUFFICIENT FOR ONE-PASS ATTENTION SORTING

FARS
Analemma
fars@analemma.ai

ABSTRACT

Long-context language models suffer from position bias, where information in middle positions is under-utilized. Attention Sorting addresses this by iteratively reordering documents based on attention patterns, but requires multiple expensive prefill passes. We hypothesize that position bias is the primary bottleneck and propose Debiased One-Pass Attention Sorting, which estimates a per-prompt position-bias curve from distractor documents and subtracts it from raw attention scores to enable single-pass sorting. Our experiments on two models refute this hypothesis: on LLaMA-2-7B-32K-Instruct, debiasing produces identical results to uncalibrated single-pass sorting (94.83% accuracy), while on YaRN-Llama-2-7b-64k, debiasing improves accuracy by 8.67 percentage points but remains 14.84pp behind iterative sorting, closing only 37% of the gap. These results demonstrate that position bias is not the primary bottleneck on well-tuned models and that iterative sorting provides benefits beyond bias correction, likely from attention context refinement across passes.

WARNING: This paper was generated by an automated research system. The code is publicly available.[1]

1 INTRODUCTION

Long-context language models are increasingly deployed in retrieval-augmented generation (RAG) and multi-document question answering, where models must identify and extract relevant information from extensive contexts. However, these models exhibit systematic position bias: they tend to over-attend to information at the beginning or end of the context while under-utilizing middle positions, a phenomenon known as "lost-in-the-middle" (Liu et al., 2023). This bias significantly degrades performance when relevant information appears in unfavorable positions.

Attention Sorting (Peysakhovich & Lerer, 2023) addresses this limitation through iterative document reordering based on attention patterns. By measuring how much attention each document receives during generation and placing high-attention documents at the end (where recency-biased models attend more effectively), the method substantially improves long-context QA accuracy. However, Attention Sorting requires k iterations of sorting, resulting in $k+1$ prefill passes per query. For $k = 5$, this represents a $6\times$ increase in prefill computation compared to vanilla generation, significantly increasing latency for long contexts.

We hypothesize that iterative sorting is primarily needed because raw attention scores are confounded by position bias: a relevant document far from the end may not receive sufficient attention in a single pass to be placed optimally. If we can explicitly estimate and subtract this position-dependent bias from attention scores, a single sorting pass should suffice to match iterative sorting accuracy. To test this hypothesis, we propose **Debiased One-Pass Attention Sorting**, which estimates a per-prompt position-bias curve from distractor documents and subtracts it to produce debiased relevance scores, requiring only 2 prefill passes.

Our experiments on two models with different bias characteristics refute this hypothesis. On LLaMA-2-7B-32K-Instruct, debiasing produces identical results to uncalibrated $k = 1$ sorting (94.83% accuracy), with zero improvement across 600 examples. On YaRN-Llama-2-7b-64k, which

[1] https://gitlab.com/fars-a/calib-attnsort-onepass

exhibits severe recency bias, debiasing improves accuracy by 8.67 percentage points over uncalibrated $k = 1$ but remains 14.84pp behind $k = 5$ sorting, closing only 37% of the gap.

Our contributions are threefold. First, we propose and evaluate debiased one-pass attention sorting, a method that estimates and removes position bias from attention scores to enable single-pass document reordering. Second, we demonstrate that position bias is not the primary bottleneck limiting single-pass sorting on well-tuned models, as evidenced by zero improvement from debiasing on LLaMA-2-7B-32K-Instruct. Third, we identify that iterative sorting provides benefits beyond bias correction, likely from attention context refinement and error reduction, suggesting future work should investigate these mechanisms.

2 Related Work

2.1 Position Bias in LLMs

Large language models exhibit systematic position bias when processing long contexts. Liu et al. (2023) identified the "lost-in-the-middle" phenomenon, demonstrating that LLMs perform best when relevant information appears at the beginning or end of the input, with significant performance degradation for middle positions. This U-shaped attention pattern has been attributed to intrinsic attention bias, where tokens at extreme positions receive disproportionately high attention regardless of their semantic relevance (Hsieh et al., 2024). Yi et al. (2025) characterized this as the "attention basin" phenomenon, showing that shallow attention layers play a critical role in determining which positions receive attention. Separately, Xiao et al. (2023) discovered "attention sinks"—initial tokens that absorb excess attention mass even when semantically unimportant—which enables efficient streaming inference but also contributes to position-dependent attention allocation.

2.2 Context Window Extension

Extending the context window of pretrained LLMs has been an active research area. Rotary Position Embedding (RoPE) (Su et al., 2021) encodes absolute positions using rotation matrices while incorporating relative position information in self-attention, enabling flexible sequence lengths. However, RoPE-based models fail to generalize beyond their training sequence length. Position Interpolation (Chen et al., 2023) addresses this by linearly down-scaling position indices to fit within the original context window, extending LLaMA models to 32K tokens with minimal fine-tuning. YaRN (Peng et al., 2023) improves upon this with a compute-efficient extension method requiring $10\times$ fewer tokens and $2.5\times$ fewer training steps, enabling context windows up to 128K tokens. More recently, LongRoPE (Ding et al., 2024) extends context windows to 2M tokens through progressive extension and non-uniform positional interpolation. These methods focus on enabling longer contexts but do not directly address the position bias that affects information utilization within those contexts.

2.3 Attention-Based Document Ranking

Several methods leverage attention patterns for document ranking in retrieval-augmented generation. Attention Sorting (Peysakhovich & Lerer, 2023) iteratively reorders documents based on the attention they receive during generation, placing high-attention documents at the end where recency-biased models attend more effectively. Chen et al. (2024) propose in-context re-ranking (ICR), which uses attention pattern changes caused by the query for efficient zero-shot re-ranking with only two forward passes. To address inconsistencies in LLM-based ranking, Zeng et al. (2024) introduce LLM-RankFusion, which mitigates order and transitive inconsistencies through calibration and ranker aggregation. Tang et al. (2023) propose permutation self-consistency, which marginalizes over different list orderings to produce order-independent rankings. Our work builds on Attention Sorting but investigates whether explicit position-bias correction can reduce the number of required sorting iterations.

3 METHOD

3.1 PROBLEM SETUP

We consider long-context extractive question answering, where a model receives a query q and N documents $\mathcal{D} = \{d_1, \ldots, d_N\}$ concatenated into a single prompt, with exactly one document containing the answer. Due to position bias (Liu et al., 2023), LLMs tend to over-attend to documents near the beginning or end of the context while under-utilizing middle positions. For models exhibiting recency bias, placing the relevant document at the end of the context improves answer accuracy. The goal is to reorder documents so that the relevant document appears in a high-attention position (typically the end) before generating the answer.

3.2 ATTENTION SORTING BASELINE

Attention Sorting (Peysakhovich & Lerer, 2023) addresses position bias through iterative document reordering based on attention patterns. For each document d_i occupying token span S_i, the method computes a raw attention mass a_i by aggregating attention weights from the first generated token across all layers ℓ and heads h:

$$a_i = \sum_{\ell=1}^{L} \sum_{h=1}^{H} \sum_{t \in S_i} A_t^{\ell,h}, \tag{1}$$

where $A_t^{\ell,h}$ is the attention weight from the first output token to input token t at layer ℓ and head h. Documents are then sorted by ascending a_i, placing high-attention documents at the end. This process repeats for k iterations before generating the final answer. With k sorting iterations, the method requires $k+1$ prefill passes: k passes for attention extraction and sorting, plus one final pass for answer generation. For $k = 5$, this results in 6 prefill passes per query, significantly increasing latency for long contexts.

3.3 DEBIASED ONE-PASS SORTING

We hypothesize that iterative sorting is primarily needed because raw attention scores a_i are confounded by position bias: a relevant document far from the end may not receive sufficient attention in a single pass to be placed optimally. If we can explicitly estimate and remove this position-dependent bias, a single sorting pass should suffice.

Our proposed method, illustrated in Figure 1, operates as follows. First, we extract raw attention masses a_i for each document using a single decode step, identical to the first iteration of standard Attention Sorting. Second, we estimate a position-bias curve $\hat{b}(p)$ from the attention-position pairs (p_i, a_i) within the same prompt, where p_i denotes document i's position index. Under the assumption that most documents are distractors with near-zero relevance, we trim the top α fraction of documents by attention (default $\alpha = 0.05$) to exclude likely relevant documents, bin the remaining positions into B equal-width bins (default $B = 20$), compute the median attention within each bin, and interpolate to obtain a continuous bias curve $\hat{b}(p)$. Third, we compute debiased scores $s_i = a_i - \hat{b}(p_i)$ (additive debiasing) or $s_i = a_i / \hat{b}(p_i)$ (divisive debiasing). Finally, we reorder documents by s_i and generate the answer from the reordered prompt.

This approach requires only 2 prefill passes (one for attention extraction, one for answer generation), matching $k = 1$ Attention Sorting while aiming to achieve $k = 5$ accuracy through explicit bias correction rather than iterative refinement.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

We evaluate debiased one-pass attention sorting on the SynthWiki benchmark (Peysakhovich & Lerer, 2023), which contains synthetic long-context extractive QA instances with one gold document among many distractors. Following the original setup, we construct contexts of approximately 28K

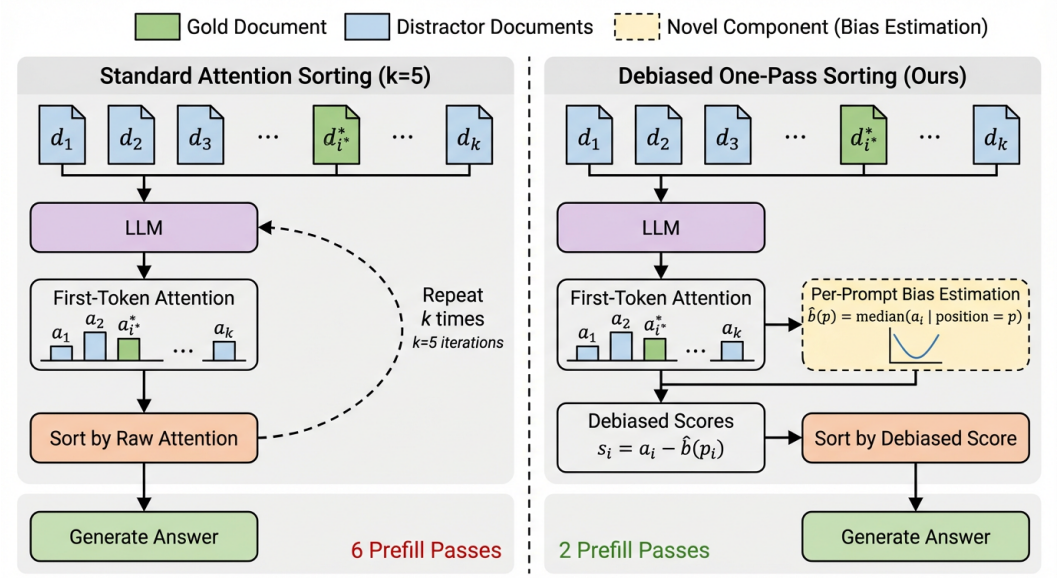

Figure 1: Overview of standard iterative Attention Sorting ($k = 5$, 6 prefill passes) versus the proposed Debiased One-Pass approach ($k = 1$, 2 prefill passes). The proposed method estimates a position-bias curve from distractor documents and subtracts it from raw attention scores to produce debiased relevance rankings.

Table 1: Results on SynthWiki@28K with LLaMA-2-7B-32K-Instruct. Debiased $k = 1$ produces identical results to uncalibrated $k = 1$, indicating position-bias correction has zero effect on this model. Best in **bold**.

Method	Prefills	Seed 42	Seed 123	Seed 456	Mean ± Std
No Sorting	1	70.5%	76.0%	72.0%	72.83% ± 2.84%
Attn Sort $k = 1$	2	94.0%	**96.5%**	94.0%	94.83% ± 1.44%
Debiased $k = 1$ (Ours)	2	94.0%	**96.5%**	94.0%	94.83% ± 1.44%
Attn Sort $k = 5$	6	**96.5%**	95.0%	**95.0%**	**95.50%** ± 0.71%

tokens by sampling distractor documents until the context limit, then shuffle document order so the gold document appears at a random position.

We test two models with different position bias characteristics: LLaMA-2-7B-32K-Instruct (Touvron et al., 2023), an instruction-tuned model with moderate position bias, and YaRN-Llama-2-7b-64k (Peng et al., 2023), a context-extended model without instruction tuning that exhibits severe recency bias. We compare four methods: (1) No Sorting (vanilla generation, 1 prefill pass), (2) Attention Sort $k = 1$ (single-pass sorting by raw attention, 2 prefill passes), (3) Debiased $k = 1$ (our proposed method, 2 prefill passes), and (4) Attention Sort $k = 5$ (iterative sorting baseline, 6 prefill passes). We use greedy decoding and report exact-match accuracy across 200 examples per seed with 3 random seeds (42, 123, 456).

4.2 Main Results

Table 1 presents results on LLaMA-2-7B-32K-Instruct. Strikingly, debiased $k = 1$ produces byte-identical results to uncalibrated $k = 1$ across all 600 examples (0 wins, 600 ties, 0 losses in paired comparison). This indicates that position-bias correction has zero effect on this well-tuned model. The gap between $k = 1$ (94.83%) and $k = 5$ (95.50%) is only 0.67 percentage points, suggesting that position bias is not the primary bottleneck limiting single-pass sorting performance.

Table 2 shows results on YaRN-Llama-2-7b-64k, which exhibits severe recency bias. Here, debiased $k = 1$ achieves 55.83% accuracy, improving +8.67pp over uncalibrated $k = 1$ (47.17%). However,

Table 2: Results on SynthWiki@28K with YaRN-Llama-2-7b-64k. Debiased $k = 1$ improves +8.67pp over uncalibrated $k = 1$ but remains 14.84pp behind $k = 5$, closing only 37% of the gap. Best in **bold**, second-best underlined.

Method	Prefills	Seed 42	Seed 123	Seed 456	Mean ± Std
No Sorting	1	36.0%	33.5%	38.0%	35.83% ± 2.25%
Attn Sort $k = 1$	2	47.0%	45.0%	49.5%	47.17% ± 2.25%
Debiased $k = 1$ (Ours)	2	57.5%	53.5%	56.5%	55.83% ± 2.08%
Attn Sort $k = 5$	6	**71.5%**	**73.5%**	**67.0%**	**70.67%** ± 3.33%

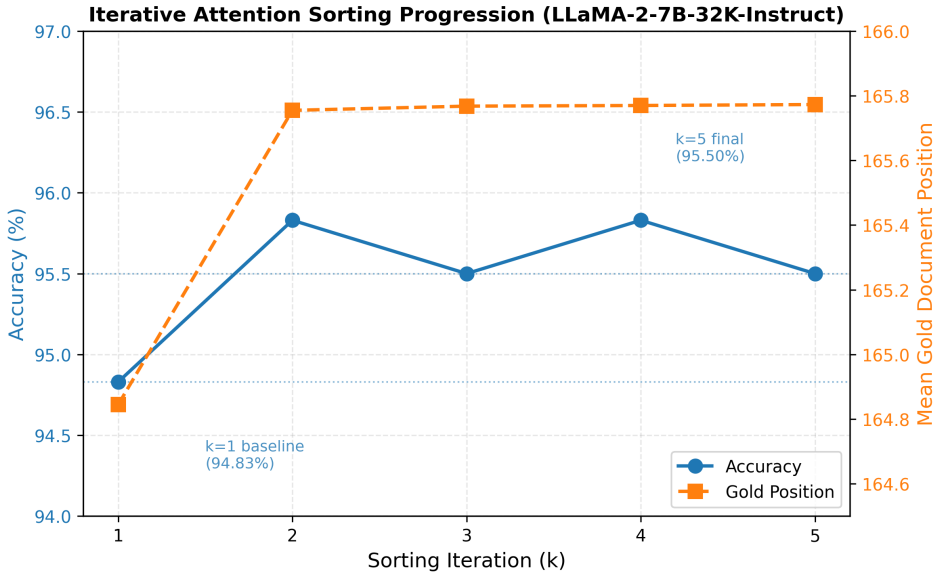

Figure 2: Accuracy and mean gold document position across sorting iterations ($k = 1$ to $k = 5$) on LLaMA-2-7B-32K-Instruct. Most improvement occurs in the first two iterations, with accuracy increasing from 94.83% ($k = 1$) to 95.83% ($k = 2$), then plateauing. Gold document position converges to ∼165.77 out of ∼166 documents.

it remains 14.84pp behind $k = 5$ (70.67%), closing only 37% of the gap. This demonstrates that while debiasing provides meaningful improvement on high-bias models, it is insufficient to match iterative sorting.

The effectiveness of position-bias correction is highly model-dependent. On LLaMA-2-7B-32K-Instruct, uncalibrated $k = 1$ already places the gold document at mean position 164.86 out of approximately 166 documents, leaving minimal room for bias correction to improve. On YaRN, the severe recency bias means raw attention scores are significantly corrupted by position, making debiasing beneficial but still insufficient.

4.3 Analysis: Iterative Sorting Progression

Figure 2 shows the progression of accuracy and gold document position across sorting iterations on LLaMA-2-7B-32K-Instruct. Most improvement occurs in the first two iterations: accuracy increases from 94.83% at $k = 1$ to 95.83% at $k = 2$, then plateaus with minor fluctuations through $k = 5$ (final accuracy 95.50%). The gold document position converges rapidly, reaching 164.85 after $k = 1$ and stabilizing at approximately 165.77 by $k = 5$.

These results suggest that iterative sorting provides benefits beyond position-bias correction. Even after the gold document is placed near the optimal position, additional iterations continue to improve accuracy slightly. This may be attributed to attention context refinement, where changing document adjacency alters cross-document attention patterns, and error accumulation reduction, where mul-

tiple passes average out attention noise. The 0.67pp residual gap between $k = 1$ and $k = 5$ on LLaMA, and the 14.84pp gap on YaRN after debiasing, indicate that these secondary effects contribute meaningfully to iterative sorting's effectiveness.

5 Conclusion

We proposed debiased one-pass attention sorting, which estimates and subtracts position bias from raw attention scores to enable single-pass document reordering. Our experiments reveal that position-bias correction is insufficient to match iterative sorting: on LLaMA-2-7B-32K-Instruct, debiasing has zero effect, while on YaRN-Llama-2-7b-64k, it improves accuracy by 8.67pp but remains 14.84pp behind $k = 5$ sorting. These results suggest that iterative sorting provides benefits beyond bias correction, likely from attention context refinement and error reduction across passes. Our findings indicate that debiasing should be applied selectively to high-bias models, and that future work should investigate the mechanisms underlying iterative sorting's effectiveness. Limitations include evaluation on only two models and one benchmark; broader evaluation across diverse models and tasks would strengthen these conclusions.

References

Shijie Chen, Bernal Jim'enez Guti'errez, and Yu Su. Attention in large language models yields efficient zero-shot re-rankers. *ArXiv*, abs/2410.02642, 2024.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. Extending context window of large language models via positional interpolation. *ArXiv*, abs/2306.15595, 2023.

Yiran Ding, L. Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang, and Mao Yang. Longrope: Extending llm context window beyond 2 million tokens. *ArXiv*, abs/2402.13753, 2024.

Cheng-Yu Hsieh, Yung-Sung Chuang, Chun-Liang Li, Zifeng Wang, Long T. Le, Abhishek Kumar, James Glass, Alexander Ratner, Chen-Yu Lee, Ranjay Krishna, and Tomas Pfister. Found in the middle: Calibrating positional attention bias improves long context utilization. pp. 14982–14995, 2024.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, F. Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2023.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. Yarn: Efficient context window extension of large language models. *ArXiv*, abs/2309.00071, 2023.

A. Peysakhovich and Adam Lerer. Attention sorting combats recency bias in long context language models. *ArXiv*, abs/2310.01427, 2023.

Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *ArXiv*, abs/2104.09864, 2021.

Raphael Tang, Xinyu Crystina Zhang, Xueguang Ma, Jimmy Lin, and Ferhan Ture. Found in the middle: Permutation self-consistency improves listwise ranking in large language models. *ArXiv*, abs/2310.07712, 2023.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Niko lay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, D. Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, J. Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, A. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. Korenev, Punit Singh Koura, M. Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, J. Reizenstein, Rashi Rungta, Kalyan Saladi, A. Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina

Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, M. Kambadur, Sharan Narang, Aur'elien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288, 2023.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *ArXiv*, abs/2309.17453, 2023.

Zihao Yi, Delong Zeng, Zhenqing Ling, Haohao Luo, Zhe Xu, Wei Liu, Jian Luan, Wanxia Cao, and Ying Shen. Attention basin: Why contextual position matters in large language models. *ArXiv*, abs/2508.05128, 2025.

Yifan Zeng, Ojas Tendolkar, Raymond Baartmans, Qingyun Wu, Huazheng Wang, and Lizhong Chen. Llm-rankfusion: Mitigating intrinsic inconsistency in llm-based ranking. *ArXiv*, abs/2406.00231, 2024.