Learning from Emptiness: De-biasing Listwise Rerankers with Content-Agnostic Probability Calibration

Hang Lv[1]*, Hongchao Gu[1]*, Ruiqing Yang[1], Liangyue Li[2], Zulong Chen[2], Defu Lian[1], Hao Wang[1]†, Enhong Chen[1]†
[1]University of Science and Technology of China [2]Alibaba Group

Abstract

Generative listwise reranking leverages global context for superior retrieval but is plagued by intrinsic position bias, where models exhibit structural sensitivity to input order independent of relevance. Existing mitigations present a dilemma: inference-time aggregation incurs prohibitive latency, while training-based methods often fail to eradicate ingrained priors, particularly in compact models. To resolve this dilemma, we propose CapCal (Content-Agnostic Probability Calibration), a training-free framework that mechanically decouples positional bias from ranking decisions. By estimating the bias distribution via content-free placeholders, CapCal rectifies output logits through an entropy-adaptive contrastive mechanism. Evaluations across 10 benchmarks confirm that CapCal achieves superior performance among training-free methods while preserving single-pass efficiency. Notably, it unlocks the latent potential of lightweight models (e.g., 0.6B), delivering absolute NDCG gains exceeding 10 points and outperforming both permutation-based aggregation and data-augmentation baselines. Our code is openly available at USTC-StarTeam/CapCal.

1 Introduction

Listwise reranking (Sun et al., 2023; Ma et al., 2023) has established itself as a powerful paradigm in modern information retrieval, utilizing Large Language Models to evaluate candidate documents simultaneously rather than in isolation. By integrating the query and the full document list within a single prompt, this approach enables the model to capture global inter-document dependencies, delivering ranking performance that substantially outperforms traditional pointwise or pairwise methods.

However, this paradigm faces a fundamental theoretical flaw: the violation of *Permutation Invariance*. Theoretically, an ideal ranking function must yield results that are invariant to the physical sequence of candidate documents. In practice, however, LLMs exhibit intrinsic sensitivity to token positions, inevitably giving rise to severe **position bias**. This often manifests as the "Lost in the Middle" phenomenon (Liu et al., 2024), where models systematically favor documents at the beginning or end of the context while overlooking highly relevant ones in the middle, thereby degrading the overall reliability of the retrieval system.

While extensive research has been dedicated to mitigating this bias, existing methodologies remain constrained by an inherent trade-off between efficiency and flexibility. On one hand, inference-time aggregation strategies (Tang et al., 2024; Lee et al., 2025) attempt to neutralize bias by averaging predictions across multiple input permutations; however, the computational overhead incurs prohibitive latency, rendering them impractical for real-time scenarios. On the other hand, training-based interventions (Pradeep et al., 2023a; Ren et al., 2024) seek to enforce order invariance through extensive data augmentation or architectural modifications. Yet, these approaches introduce significant rigidity: they necessitate costly retraining for any new backbone and often exhibit suboptimal cross-domain generalization. Consequently, there remains a critical need for an efficient, training-free solution capable of rectifying position bias without succumbing to the computational burden of aggregation or the inflexibility of retraining.

In this work, we propose CapCal (**C**ontent-**A**gnostic **P**robability **Cal**ibration), a novel framework that resolves this dilemma by fundamentally reconceptualizing the nature of position bias. Rather than treating bias merely as a symptom to be suppressed, we formulate it as a measurable component to be explicitly decoupled. As illus-

*Equal contribution. Work partially done during an internship at Alibaba Group.
†Corresponding authors. Contact wanghao3@ustc.edu.cn

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 817–826

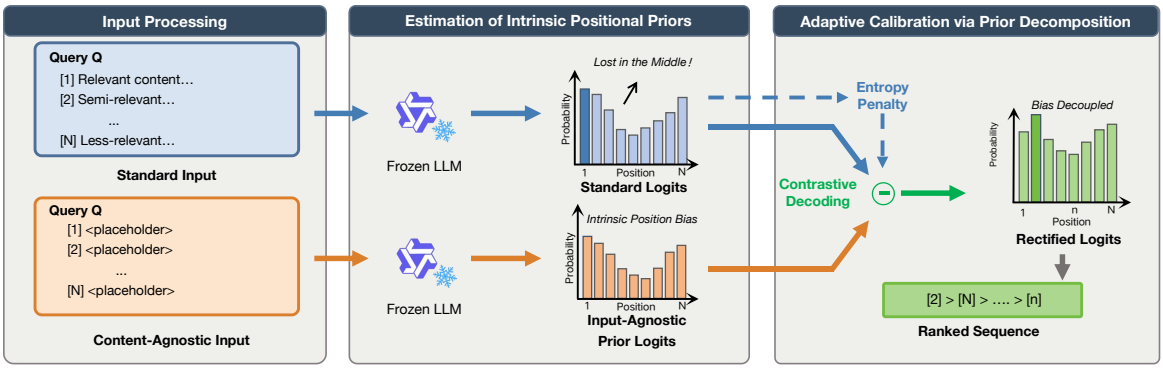

Figure 1: **Overview of the proposed CapCal framework.** The framework decouples position bias by utilizing an empty-passage query to capture the input-agnostic prior. We then apply contrastive decoding to subtract this prior from the standard inference logits, achieving a calibrated ranking.

trated in Figure 1, our approach is grounded in the key insight that intrinsic priors can be isolated by observing the model's behavior in the absence of content. Specifically, by querying the model with empty passages, we capture the content-agnostic prior—a baseline preference for specific list indices manifested directly in the logits. We then leverage contrastive calibration to analytically subtract this prior from the standard generation probabilities. This plug-and-play strategy rectifies ranking distributions without sacrificing listwise inference efficiency. We validate CapCal across 10 benchmarks (MS MARCO and BEIR) using Qwen models ranging from 0.6B to 8B. Empirical results demonstrate that our framework consistently enhances ranking performance while maintaining single-pass efficiency. Notably, it unlocks the potential of smaller models, delivering over 10 NDCG points of improvement in high-bias scenarios and proving robust across diverse retrieval contexts. Our contributions are summarized as follows:

- We introduce CapCal, a plug-and-play framework that mechanically rectifies position bias via content-agnostic probability calibration, eliminating the need for retraining or expensive inference aggregation.
- We bridge the capability gap for small language models, demonstrating that explicit bias correction allows 0.6B-scale models to rival larger baselines, thereby enhancing the scalability of generative reranking.
- We reveal the limitations of implicit adaptation, showing that inference-time calibration is superior to permutation-based methods in handling the stubborn, intrinsic nature of positional priors.

2 Related Work

2.1 Generative listwise reranking

Generative listwise reranking has superseded pointwise and pairwise paradigms by enabling LLMs to capture global inter-document dependencies. Sun et al. (2023) and Ma et al. (2023) pioneered this approach by prompting LLMs to directly generate document permutations. Subsequent efforts, such as Pradeep et al. (2023a) and Pradeep et al. (2023b), optimized this via instruction fine-tuning on open-source backbones, while Gangi Reddy et al. (2024) further reduced latency by deriving rankings solely from the first token's logits. Although recent trends explore integrating reasoning steps to enhance precision(Liu et al., 2025; Weller et al., 2025; Yang et al., 2025b), such methods incur severe latency costs due to extensive token generation. Consequently, direct generation with efficient, smaller-scale models remains a practical choice for latency-sensitive applications, making robust position-bias mitigation especially important.

2.2 Position Bias and Mitigation Strategies

LLMs exhibit intrinsic sensitivity to input order, leading to the "Lost in the Middle" phenomenon (Liu et al., 2024). Mitigation strategies primarily diverge into inference-time input manipulation and model-level adaptation. **Inference-time approaches** treat the LLM as a black box. Aggregation strategies, such as Permutation Self-Consistency (PSC) (Tang et al., 2024), LLM-RankFusion (Zeng et al., 2024), and Mixture-of-Intervention (MOI) (Lee et al., 2025), neutralize positional noise by averaging or solving for rankings across multiple shuffled inputs. Decomposition

strategies like sliding windows (Sun et al., 2023) or tournament sorting (Yoon et al., 2024) mitigate bias by segmenting lists to limit context length. While effective, these methods incur prohibitive latency due to repeated inference. Conversely, **model-level approaches** target internal mechanisms. Data augmentation implies training on shuffled samples to learn invariance(Pradeep et al., 2023a,b), while architectural modifications—such as ListT5's (Yoon et al., 2024) ranking loss or SCaLR's (Ren et al., 2024) dual-view alignment—enforce it structurally. However, these techniques often suffer from inflexibility and potential generalization issues due to costly retraining. Lastly, calibration methods like ICR (Chen et al., 2025) and FitM (Hsieh et al., 2024) leverage internal attention weights to mathematically decouple positional priors. While sharing the goal of parameter-free bias isolation, these methods primarily focus on normalizing attention maps rather than rectifying generative distribution.

2.3 Decoding Strategies and Calibration

Contrastive Decoding (CD) (Li et al., 2023; O'Brien and Lewis, 2023) was originally proposed to enhance generation quality by maximizing the divergence between an "expert" and an "amateur" model, effectively suppressing generic or repetitive modes. To eliminate the need for auxiliary models, self-contrastive approaches (Chuang et al., 2024) derive the negative constraint from the model itself, typically via unconditional prompts(Peng et al., 2025) or early-exit layers(Phan et al., 2024). In Retrieval-Augmented Generation, methods like Context-Aware Decoding (CAD) (Shi et al., 2024; Zhao et al., 2024; Qiu et al., 2025) apply this principle to amplify contextual reliance by subtracting context-agnostic priors. Recent decoding-time steering methods further show that generation behavior can be adjusted at inference time without updating model parameters, for example through collaborative logit steering or speculative-style personalized generation (Lv et al., 2026a,b). Our work extends this idea to listwise reranking: we treat the empty-context prior as the mathematical manifestation of positional bias and subtract it from the ranking distribution, yielding a training-free debiasing mechanism tailored to generative ranking.

3 Methodology

We present **CapCal**, a training-free framework designed to calibrate listwise reranking by explicitly decoupling inherent positional priors from semantic relevance. As illustrated in Figure 1, our approach operates in two stages: quantifying the input-agnostic prior via semantic-free placeholders, and rectifying the ranking distribution through an adaptive, contrastive calibration mechanism.

3.1 Preliminaries: Listwise Reranking

Given a query q and a list of N candidate documents $\mathcal{D} = \{d_1, d_2, \ldots, d_N\}$, the goal is to generate a permutation π of indices that ranks documents by decreasing relevance. Following the standard generative paradigm (Sun et al., 2023), the input prompt x is constructed by concatenating the query and the document list: $x = \mathcal{T}(q, d_1, \ldots, d_N)$, where $\mathcal{T}$ is a prompt template as shown in C. At step k, the probability of generating the identifier token t_k corresponding to document d_i is computed as: $P(t_k = d_i \mid x, t_{<k}) = \frac{\exp(\mathbf{z}_{k,d_i})}{\sum_{v \in \mathcal{V}} \exp(\mathbf{z}k,v)}$, where $\mathbf{z}_k \in \mathbb{R}^{|\mathcal{V}|}$ are the logits over the vocabulary V.

3.2 Estimation of Intrinsic Positional Priors

Position bias intrinsically manifests as a non-uniform probability distribution over document indices, persisting independently of semantic content. To isolate this bias, we construct a content-free input x_{empty}. Specifically, we mirror the standard prompt structure but replace the textual content of each candidate document with a semantically vacuous $\varnothing$ (e.g., an empty string), while retaining the original query and document identifiers:

$$x_{\text{empty}} = \mathcal{T}(q, \varnothing, \ldots, \varnothing) \tag{1}$$

By feeding x_{empty} into the frozen LLM, we obtain the prior logits $\mathbf{z}_k^{\text{prior}}$. Since the candidate list is devoid of semantic information, any divergence in the probability mass assigned to different document indices (e.g., a structural preference for "1" over "5") serves as a direct quantification of the model's intrinsic positional bias.

3.3 Adaptive Calibration via Prior Decomposition

Standard contrastive decoding methods operate on token-level logits, which is ill-suited for listwise reranking where document identifiers vary in token length (e.g., "9" vs. "10"). Consequently, we formulate our calibration strategy directly within the **probability space**, focusing on alignment, decomposition, and adaptive correction.

Identifier-Level Probability Alignment. First, we estimate the generation probability for each candidate document d_i. To normalize across identifiers of varying lengths, we compute the joint probability of the constituent token sequence $(w_1, \ldots, w_m)$ appended with a termination symbol (e.g., '⌋') to normalize across lengths:

$$P(d_i \mid x) = \prod_{j=1}^{m} P(w_j \mid x, w_{<j}) \cdot P(\rfloor \mid x, w_{1:m}) \tag{2}$$

Decomposition of Positional Bias. We define the calibrated score $S(d_i)$ by decomposing the model's prediction into a semantic component and a bias component. Theoretically, in the complete absence of semantic signals, an ideal unbiased ranker must yield a uniform distribution over the candidate set. Deviations from this uniform baseline constitute excessive prior bias. The calibrated score is derived by subtracting this bias from the original probability:

$$S(d_i) = P(d_i \mid x) - \alpha \cdot \left(P(d_i \mid x_{\text{empty}}) - \frac{1}{|\mathcal{C}_k|} \right) \tag{3}$$

where $\mathcal{C}_k$ denotes the set of available candidates at step k, and $\frac{1}{|\mathcal{C}_k|}$ represents the theoretical uniform assumption.

Entropy-Based Adaptive Penalty. Instead of a static penalty, we introduce a dynamic coefficient α_k to adapt to the model's fluctuating uncertainty. This dynamic adjustment is motivated by the observation that LLMs tend to fall back on positional shortcuts primarily when semantic confidence diminishes. Accordingly, we modulate α_k based on the Shannon entropy of the prediction distribution:

$$\alpha_k = \beta \cdot \mathcal{H}(P(\cdot \mid x, t_{<k})) = -\beta \sum_{d \in \mathcal{C}_k} p(d) \log p(d) \tag{4}$$

where $p(d)$ is the normalized probability of candidate d, and β is a scaling hyperparameter. This ensures strong calibration when uncertainty is high (flat distribution), while preserving original semantics when the model is confident.

Constrained Decoding. Finally, to guarantee the validity of the output ranking, we employ constrained decoding. At each step, the vocabulary is restricted to the valid next tokens corresponding to indices in $\mathcal{C}_k$, ensuring the generated sequence forms a valid permutation of the input documents.

4 Experiments

4.1 Experiment Setup

Tasks. We conduct a comprehensive evaluation on 10 benchmarks spanning general web search and specialized domains. For *General Web Search*, we adhere to the standard MS MARCO (Bajaj et al., 2018) protocols, covering the v1 corpus with TREC DL19 and DL20 (Craswell et al., 2020, 2021) and the v2 corpus with TREC DL21–DL23 (Craswell et al., 2025a,b,c).[1] For *Specialized Domains*, we adopt a diversity-first selection strategy over BEIR, choosing five datasets from distinct semantic fields: Epidemiology (TREC-COVID), Nutritional Science (NFCorpus), Earth Science (Climate-FEVER), Finance (FiQA), and Argumentation (Arguana). This yields a balanced suite of five general-domain and five specialized-domain benchmarks.

Models. To assess scalability and robustness, we evaluate models of different sizes and generations: Qwen3-0.6B, Qwen3-8B (Yang et al., 2025a), and Qwen2.5-7B-Instruct (Qwen et al., 2025), a widely used instruction-tuned baseline.

4.2 Main Results

Table 1 presents the comparative analysis of our framework across three distinct model scales and ten diverse benchmarks, together with in-domain and out-of-domain averages for quick comparison. The empirical evidence demonstrates that our training-free calibration mechanism consistently enhances generative ranking performance across almost all settings.

Universal Efficacy Across Scales. Our framework yields robust gains across all tested architectures, ranging from standard 7B/8B models to the highly compact 0.6B variant. While effective universally, the impact is particularly pronounced on Qwen3-0.6B, achieving absolute improvements exceeding 10 NDCG@10 points on MS MARCO V2 and Climate-Fever. This indicates that while bias rectification benefits models of all sizes, it is instrumental in unlocking the latent ranking potential of smaller models, which are intrinsically more susceptible to positional sensitivity.

[1] One reviewer pointed out that the evaluation on TREC DL21 may be less reliable; we therefore retain its results for completeness but treat them as reference only.

		MSv1		MSv2			Avg.	BEIR					Avg.
Model	**Method**	**DL19**	**DL20**	**DL21**	**DL22**	**DL23**	**InD**	**COVID**	**Clim.**	**NF**	**Argu.**	**FiQA**	**OOD**
Qwen3-0.6b	Base	0.4916	0.3415	0.6278	0.4858	0.5457	0.4985	0.6074	0.1622	0.4243	0.1097	0.1807	0.2969
	CapCal	**0.5454**	**0.4126**	**0.6666**	**0.5457**	**0.5762**	**0.5493**	**0.6400**	**0.2086**	**0.4504**	**0.2448**	**0.2364**	**0.3560**
Qwen3-8b	Base	0.6807	0.5405	0.8091	0.6129	0.6945	0.6675	0.7525	0.2486	0.4981	0.2436	0.2877	0.4061
	CapCal	**0.7141**	**0.5613**	**0.8228**	**0.6391**	**0.6979**	**0.6870**	**0.7683**	**0.2714**	**0.5094**	**0.2464**	**0.2932**	**0.4177**
Qwen2.5-7b	Base	**0.7114**	**0.5151**	0.8142	0.6293	0.6317	**0.6603**	0.7542	0.2325	0.4858	0.2181	0.2846	0.3950
	CapCal	0.6991	0.5020	**0.8277**	**0.6417**	**0.6411**	0.6623	**0.7665**	**0.2405**	**0.5088**	**0.2553**	**0.3075**	**0.4157**

Table 1: Performance comparison on MS MARCO and BEIR benchmarks. All metrics are reported in NDCG@10.

Cross-Domain Generalization. The method exhibits remarkable stability across both general web search and specialized domains. By explicitly decoupling intrinsic positional priors, our calibration restores ranking fidelity, effectively handling zero-shot scenarios and complex reasoning tasks such as TREC DL23 and NFCorpus. The sustained improvements across both MS MARCO settings and diverse BEIR tasks further attest to the domain-agnostic nature of our adaptive penalty mechanism.

Comparison with Inference-Time Aggregation. We further compare CapCal against Permutation Self-Consistency (PSC), a strong inference-time aggregation baseline that performs 10 shuffled reranking passes per query. As detailed in Appendix A.4, CapCal consistently matches or exceeds PSC on Qwen3-0.6B while requiring only one additional forward pass, demonstrating a substantially better effectiveness-efficiency trade-off.

4.3 Discussion and Analysis

Using Qwen3-0.6B, we analyze CapCal's robustness and examine whether training-based debiasing can substitute for inference-time calibration.

Robustness Analysis. We challenge our calibration mechanism under three perturbation settings to verify its stability:

- **Input Distribution Shift:** We altered the input distribution by employing a stronger first-stage retriever (*bge-reranker-v2-m3* (Chen et al., 2024)) and applying *Random Shuffling* to the candidate list. Our method yields consistent gains regardless of initial retrieval quality or randomized ordering, demonstrating its robustness to input variations and its orthogonality to document sequence. (Appendix A.1)
- **Placeholder Sensitivity:** We varied both the semantic content and the length of the placeholders used to capture priors. The calibration remains effective across these variations, confirming that the captured bias is structural—stemming from the position indices themselves—rather than being triggered by specific semantic tokens in the dummy input. (Appendix A.2)
- **Identifier Semantics:** We shifted document identifiers from standard numerals (e.g., "1") to alphabets (e.g., "A"). The framework successfully generalizes to these alternative formats, demonstrating adaptability to diverse prompting strategies. (Appendix A.3)

Persistence of Bias in Data Augmentation. Seminal works such as RankZephyr (Pradeep et al., 2023b) uses permutation-based data augmentation to suppress position bias. We replicate this strategy by fine-tuning Qwen3-0.6B, but find that substantial positional priors remain. This suggests that compact models may not fully erase such biases through data diversity alone, whereas CapCal explicitly corrects the residual bias at inference time. (Appendix A.5)

5 Conclusion

We present **CapCal**, a training-free framework that mitigates position bias in generative listwise reranking by estimating content-agnostic priors and calibrating ranking probabilities. Across 10 benchmarks, CapCal improves ranking performance with single-pass efficiency, notably benefiting 0.6B models and outperforming both aggregation- and augmentation-based baselines. Future work will extend this idea to generation-oriented scenarios such as retrieval-augmented long-form generation and data-utility-aware alignment (Gu et al., 2025; Zhi et al., 2026), as well as multimodal reranking over heterogeneous inputs.

Limitations

We acknowledge that CapCal involves two primary limitations. First, the framework introduces a marginal increase in computational overhead because it requires an additional forward pass to isolate the input-agnostic prior using content-free placeholders. While this remains significantly more efficient than permutation-based aggregation strategies, it is inherently slower than standard single-pass generative reranking. Second, because our calibration is performed directly in the probability space, the method requires access to identifier-level probabilities or logits. Consequently, CapCal is not applicable to completely black-box APIs that expose only raw text completions, although it remains compatible with deployment settings that provide log-probabilities or output distributions.

Ethical Considerations

This work studies position bias in generative listwise reranking and proposes a training-free calibration method to reduce the influence of candidate order on ranking decisions. Since reranking systems may affect the visibility of information in search, recommendation, and retrieval-augmented applications, mitigating structural biases such as positional preference can contribute to fairer and more reliable information access. However, CapCal only addresses position-induced artifacts in the model's output distribution and does not eliminate other sources of bias, such as dataset bias, relevance-annotation bias, or social bias inherited from the underlying language model. In real-world deployments, calibrated rerankers should therefore be combined with task-specific fairness, safety, and quality audits. All experiments in this paper are conducted on public retrieval benchmarks, and no private user data or human-subject data are involved.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Nos. U23A20319, 62441239, 62441227, and 62472394), the Anhui Province Science and Technology Innovation Project (Nos. 202423k09020010 and 202423k09020011), and Alibaba Group through the Alibaba Innovative Research Program. The authors sincerely thank the ACL Rolling Review reviewers for their valuable suggestions.

References

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2018. Ms marco: A human generated machine reading comprehension dataset. *Preprint*, arXiv:1611.09268.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *Preprint*, arXiv:2402.03216.

Shijie Chen, Bernal Jiménez Gutiérrez, and Yu Su. 2025. Attention in large language models yields efficient zero-shot re-rankers. *Preprint*, arXiv:2410.02642.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James Glass, and Pengcheng He. 2024. Dola: Decoding by contrasting layers improves factuality in large language models. *Preprint*, arXiv:2309.03883.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2021. Overview of the trec 2020 deep learning track. *Preprint*, arXiv:2102.07662.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Jimmy Lin. 2025a. Overview of the trec 2021 deep learning track. *Preprint*, arXiv:2507.08191.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, Jimmy Lin, Ellen M. Voorhees, and Ian Soboroff. 2025b. Overview of the trec 2022 deep learning track. *Preprint*, arXiv:2507.10865.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. 2020. Overview of the trec 2019 deep learning track. *Preprint*, arXiv:2003.07820.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Hossein A. Rahmani, Daniel Campos, Jimmy Lin, Ellen M. Voorhees, and Ian Soboroff. 2025c. Overview of the trec 2023 deep learning track. *Preprint*, arXiv:2507.08890.

Revanth Gangi Reddy, JaeHyeok Doo, Yifei Xu, Md Arafat Sultan, Deevya Swain, Avirup Sil, and Heng Ji. 2024. FIRST: Faster improved listwise reranking with single token decoding. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8642–8652, Miami, Florida, USA. Association for Computational Linguistics.

Hongchao Gu, Dexun Li, Kuicai Dong, Hao Zhang, Hang Lv, Hao Wang, Defu Lian, Yong Liu, and Enhong Chen. 2025. Rapid: Efficient retrieval-augmented long text generation with writing planning and information discovery. *Preprint*, arXiv:2503.00751.

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. 2024. Found in the middle: Calibrating positional attention bias improves long context utilization. *Preprint*, arXiv:2406.16008.

Youngwon Lee, Seung-won Hwang, Daniel F Campos, Filip Graliński, Zhewei Yao, and Yuxiong He. 2025. Inference scaling for bridging retrieval and augmented generation. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 7324–7339, Albuquerque, New Mexico. Association for Computational Linguistics.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023. Contrastive decoding: Open-ended text generation as optimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12286–12312, Toronto, Canada. Association for Computational Linguistics.

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.

Wenhan Liu, Xinyu Ma, Weiwei Sun, Yutao Zhu, Yuchen Li, Dawei Yin, and Zhicheng Dou. 2025. Reasonrank: Empowering passage ranking with strong reasoning ability. *Preprint*, arXiv:2508.07050.

Hang Lv, Sheng Liang, Hao Wang, Hongchao Gu, Yaxiong Wu, Wei Guo, Defu Lian, Yong Liu, and Enhong Chen. 2026a. Costeer: Collaborative decoding-time personalization via local delta steering. *Preprint*, arXiv:2507.04756.

Hang Lv, Sheng Liang, Hao Wang, Yongyue Zhang, Hongchao Gu, Wei Guo, Defu Lian, Yong Liu, and Enhong Chen. 2026b. Specsteer: Synergizing local context and global reasoning for efficient personalized generation. *Preprint*, arXiv:2603.16219.

Xueguang Ma, Xinyu Zhang, Ronak Pradeep, and Jimmy Lin. 2023. Zero-shot listwise document reranking with a large language model. *Preprint*, arXiv:2305.02156.

Sean O'Brien and Mike Lewis. 2023. Contrastive decoding improves reasoning in large language models. *Preprint*, arXiv:2309.09117.

Keqin Peng, Liang Ding, Yuanxin Ouyang, Meng Fang, Yancheng Yuan, and Dacheng Tao. 2025. Enhancing input-label mapping in in-context learning with contrastive decoding. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 997–1004, Vienna, Austria. Association for Computational Linguistics.

Phuc Phan, Hieu Tran, and Long Phan. 2024. Distillation contrastive decoding: Improving llms reasoning with contrastive decoding and distillation. *Preprint*, arXiv:2402.14874.

Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. 2023a. RankVicuna: Zero-shot listwise document reranking with open-source large language models. *arXiv:2309.15088*.

Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. 2023b. RankZephyr: Effective and robust zero-shot listwise reranking is a breeze! *arXiv:2312.02724*.

Zexuan Qiu, Zijing Ou, Bin Wu, Jingjing Li, Aiwei Liu, and Irwin King. 2025. Entropy-based decoding for retrieval-augmented large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4616–4627, Albuquerque, New Mexico. Association for Computational Linguistics.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Ruiyang Ren, Yuhao Wang, Kun Zhou, Wayne Xin Zhao, Wenjie Wang, Jing Liu, Ji-Rong Wen, and Tat-Seng Chua. 2024. Self-calibrated listwise reranking with large language models. *Preprint*, arXiv:2411.04602.

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Wen-tau Yih. 2024. Trusting your evidence: Hallucinate less with context-aware decoding. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 783–791, Mexico City, Mexico. Association for Computational Linguistics.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Is ChatGPT good at search? investigating large language models as re-ranking agents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14918–14937, Singapore. Association for Computational Linguistics.

Raphael Tang, Crystina Zhang, Xueguang Ma, Jimmy Lin, and Ferhan Ture. 2024. Found in the middle: Permutation self-consistency improves listwise ranking in large language models. 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 2327–2340, Mexico City, Mexico. Association for Computational Linguistics.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. 2023. Zephyr: Direct distillation of lm alignment. *Preprint*, arXiv:2310.16944.

Orion Weller, Kathryn Ricci, Eugene Yang, Andrew Yates, Dawn Lawrie, and Benjamin Van Durme. 2025. Rank1: Test-time compute for reranking in information retrieval. *Preprint*, arXiv:2502.18418.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025a. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Eugene Yang, Andrew Yates, Kathryn Ricci, Orion Weller, Vivek Chari, Benjamin Van Durme, and Dawn Lawrie. 2025b. Rank-k: Test-time reasoning for listwise reranking. *Preprint*, arXiv:2505.14432.

Soyoung Yoon, Eunbi Choi, Jiyeon Kim, Hyeongu Yun, Yireun Kim, and Seung-won Hwang. 2024. ListT5: Listwise reranking with fusion-in-decoder improves zero-shot retrieval. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2287–2308, Bangkok, Thailand. Association for Computational Linguistics.

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

Zheng Zhao, Emilio Monti, Jens Lehmann, and Haytham Assem. 2024. Enhancing contextual understanding in large language models through contrastive decoding. 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 4225–4237, Mexico City, Mexico. Association for Computational Linguistics.

Xuyang Zhi, Peilun zhou, Chengqiang Lu, Hang Lv, Yiwei Liang, Rongyang Zhang, Yan Gao, YI WU, Yao Hu, Hongchao Gu, Defu Lian, Hao Wang, and Enhong Chen. 2026. Spard: Self-paced curriculum for rl alignment via integrating reward dynamics and data utility. *Preprint*, arXiv:2604.07837.

A Detailed Discussion and Analysis

A.1 Robustness to First-Stage Retrieval and Input Ordering

To evaluate the stability of our calibration mechanism, we evaluate its performance under varying initial retrieval qualities and document orderings using **Qwen3-0.6B**. We compare three distinct settings: standard **BM25** retrieval, **Random Shuffling** of BM25 results to simulate a worst-case positional bias scenario, and the high-performance **BGE-M3** reranker (Chen et al., 2024) to test the method's ceiling.

As shown in Table 2, our calibration provides consistent and significant gains regardless of the input quality. Notably, in the **Random** setting, where the base model's performance drops due to the loss of semantic ordering, our method recovers and even surpasses the standard BM25 baseline. This demonstrates that the framework effectively decouples ranking from inherent positional artifacts in the input.

Setting	Method	DL19	DL20	DL21	DL22	DL23
BM25	Base	49.16	34.15	62.78	48.58	54.57
	CapCal	**54.54**	**41.26**	**66.66**	**54.57**	**57.62**
Random	Base	47.61	32.27	59.07	47.26	43.60
	CapCal	**56.33**	**37.57**	**65.93**	**53.16**	**52.63**
BGE-M3	Base	52.32	45.23	72.06	59.07	59.68
	CapCal	**62.05**	**47.21**	**75.32**	**65.24**	**64.23**

Table 2: Detailed results of retrieval and ordering robustness on Qwen3-0.6B.

A.2 Robustness to Placeholder Content

We investigate whether the semantic content and length of the placeholder $\varnothing$ affect the estimation of positional priors. Our default **Fixed String** is the constant string "This is a placeholder", whose length is not matched to the original documents. We further compare empty, length-controlled, random, and passage-copy variants to assess the sensitivity of the prior estimate.

The results in Table 3 show that our calibration yields improvements over the uncalibrated baseline across all variations. Overall, the results confirm that the captured prior is primarily structural rather than dependent on a specific placeholder token. We also observe that the fixed string performs particularly well on DL22 and DL23, which we conjecture is because a short but non-empty buffer helps preserve clearer separation between document identifiers than a single blank space.

Placeholder Content	DL19	DL20	DL21	DL22	DL23
Base (No Calibration)	49.16	34.15	62.78	48.58	54.57
Fixed String	54.54	41.26	66.66	54.57	57.62
Passage[1] Content	56.12	40.78	64.05	50.31	52.98
Empty (a space)	55.82	44.77	63.50	50.06	50.48
Space $\times 20$	57.97	40.43	64.12	52.96	53.28
Random $\times 20$	54.15	38.92	64.58	49.03	51.43
Space $\times$len[1]	58.38	43.30	67.30	52.45	53.65
Random $\times$len[1]	50.24	40.56	65.10	51.97	49.29
Space $\times$len[i]	55.95	42.57	66.29	52.87	55.86

Table 3: Impact of different placeholder content and length on calibration performance using Qwen3-0.6B.

A.3 Robustness to Identifier Formats

To determine if the positional bias is a fundamental structural tendency, we switched document identifiers from numerals to alphabetic indices (e.g., "A", "B"). As detailed in Table 4, our calibration mechanism continues to yield clear improvements across all benchmarks with this altered format. For instance, on DL22, calibration improves the NDCG@10 from 45.60 to 53.35. This confirms that the captured positional prior is independent of specific identifier tokens, representing a generalized structural bias in the model's instruction-following behavior.

Method	DL19	DL20	DL21	DL22	DL23
Base	49.87	33.50	54.05	45.60	40.96
CapCal	**52.16**	**36.01**	**67.70**	**53.35**	**46.80**

Table 4: Robustness results using alphabetical identifiers (A, B, C...) on Qwen3-0.6B.

A.4 Comparison with PSC

We compare CapCal with Permutation Self-Consistency (PSC) (Tang et al., 2024), a strong inference-time aggregation baseline that averages rankings across 10 shuffled prompts. Table 5 shows that CapCal consistently matches or outperforms PSC on Qwen3-0.6B while using only one additional forward pass instead of 10 full reranking calls. On Qwen3-8B, the two methods are highly competitive, but CapCal remains better on DL20, DL22, DL23, Climate-FEVER, NFCorpus, and FiQA. These results indicate that directly estimating and subtracting the structural prior is a more efficient way to remove positional bias than repeatedly marginalizing over prompt orders.

Model	Method	DL19	DL20	DL21	DL22	DL23	COVID	Clim.	NF	Argu.	FiQA
Qwen3-0.6b	Base	0.4916	0.3415	0.6278	0.4858	0.5457	0.6074	0.1622	0.4243	0.1097	0.1807
	PSC	0.5382	0.4074	0.6937	0.5246	0.5264	0.5842	0.1892	0.4047	0.1910	0.1637
	CapCal	**0.5454**	**0.4126**	0.6666	**0.5457**	**0.5762**	**0.6400**	**0.2086**	**0.4504**	**0.2448**	**0.2364**
Qwen3-8b	Base	0.6807	0.5405	0.8091	0.6129	0.6945	0.7525	0.2486	0.4981	0.2436	0.2877
	PSC	**0.7404**	0.5429	**0.8473**	0.6244	0.6948	**0.7753**	0.2627	0.4980	**0.2527**	0.2856
	CapCal	0.7141	**0.5613**	0.8228	**0.6391**	**0.6979**	0.7683	**0.2714**	**0.5094**	0.2464	**0.2932**

Table 5: Comparison with Permutation Self-Consistency (PSC, $k = 10$). All metrics are reported in NDCG@10.

A.5 Limitations of Training-based De-biasing

We further investigate whether supervised fine-tuning (SFT) using permutation-augmented data can substitute for inference-time calibration. We compare *Zephyr* (Tunstall et al., 2023) and its augmented version *RankZephyr* (Pradeep et al., 2023b) against both standard and rerank-trained *Qwen3-0.6B* models.

The results in Table 6 demonstrate that training-based de-biasing is not a universal solution:

- **Persistence of Bias**: While rerank-training improves absolute scores for Qwen3-0.6B, the fine-tuned base model still exhibits substantial positional priors that our calibration can further rectify (e.g., DL19 improves from 61.15 to 65.53). This indicates that data augmentation alone cannot fully erase "hard-coded" biases in compact models.
- **Limits of Augmentation**: Even when training on large-scale augmented datasets, the residual bias in smaller architectures remains problematic. Unlike RankZephyr which shows saturation at a larger scale, compact models consistently benefit from explicit inference-time calibration as a necessary complement to standard training-based interventions.

Model	Method	DL19	DL20	DL21	DL22	DL23
Zephyr	Base	48.50	36.91	58.20	51.44	46.72
	CapCal	**51.85**	36.61	**62.83**	**53.81**	**47.90**
RankZephyr	Base	**76.67**	60.73	**89.20**	**75.11**	**76.78**
	CapCal	76.25	**60.96**	89.15	74.58	75.55
Qwen3-0.6B	Base	49.16	34.15	62.78	48.58	54.57
	CapCal	**54.54**	**41.26**	**66.66**	**54.57**	**57.62**
Qwen3-0.6B (rerank trained)	Base	61.15	43.26	78.96	65.68	66.80
	CapCal	**65.53**	**47.58**	**80.48**	**68.15**	**69.21**

Table 6: Comparative analysis between standard models and those fine-tuned with permutation-based data augmentation.

B Ranking Prompt

```
<|system|>
You are RankLLM, an intelligent assistant that can rank passages based on their relevancy to the query.
<|user|>
I will provide you with {num} passages, each indicated by a numerical identifier []. Rank the passages based on their relevance to the search query: {query}.

[1] {passage 1}
[2] {passage 2}
...
[{num}] {passage {num}}

Search Query: {query}.

Rank the {num} passages above based on their relevance to the search query. All the passages should be included and listed using identifiers, in descending order of relevance. The output format should be [] > [], e.g., [4] > [2]. Only respond with the ranking results, do not say any word or explain.
<|assistant|>


Model Generation: [9] > [4] > [20] > ... > [13]
```

Figure 2: The reranking prompt and sample generation for listwise reranking.

C AI Assistance Statement

We used AI tools only for language polishing and minor stylistic refinement. The research idea, method design, code implementation, experimental analysis, and initial manuscript drafting were completed by the authors.