ProRank: Prompt Warmup via Reinforcement Learning for Small Language Models Reranking

Xianming Li [2†*], **Aamir Shakir** [1†], **Rui Huang** [1],
Julius Lipp [1], **Benjamin Clavié** [1], **Jing Li** [2]
[1] Mixedbread AI, Berlin, Germany
[2] The Hong Kong Polytechnic University, Hong Kong SAR
{sean,aamir,rui,julius,ben}@mixedbread.ai
xianming.li@connect.polyu.hk, jing-amelia.li@polyu.edu.hk
GitHub [‡] HuggingFace [§]

Abstract

Reranking is fundamental to information retrieval and retrieval-augmented generation, with recent Large Language Models (LLMs) significantly advancing reranking quality. Most current works rely on large-scale LLMs (>7B parameters), presenting high computational costs. Small Language Models (SLMs) offer a promising alternative because of computational efficiency. However, our preliminary quantitative analysis reveals key limitations of SLMs: their representation space is narrow, leading to reduced expressiveness, and they struggle with understanding task prompts without fine-tuning. To address these issues, we introduce a novel two-stage training approach, **ProRank**, for SLM-based document reranking. We propose using reinforcement learning to improve the understanding of task prompts. Additionally, we introduce fine-grained score learning to enhance representation expressiveness and further improve document reranking quality. Extensive experiments suggest that ProRank consistently outperforms both the most advanced open-source and proprietary reranking models. Notably, our 0.5B ProRank even surpasses powerful LLM reranking models on BEIR benchmark, establishing that properly trained SLMs can achieve superior document reranking performance while maintaining computational efficiency.

1 Introduction

Document reranking is crucial in information retrieval and retrieval-augmented generation, which aims to reorder document lists initially retrieved by retrievers like BM25 based on query-document relevance (Zhu et al., 2023). Very recently, Large Language Models (LLMs) have demonstrated remarkable performance in document reranking tasks (Ma et al., 2023; Sun et al., 2023; Zhuang et al., 2023, 2024; Sun et al., 2025; Zhuang et al., 2025; Weller et al., 2025), establishing a significant new direction in document reranking research. Recent approaches have primarily employed LLMs through zero- or few-shot prompt engineering (Sun et al., 2023; Zhuang et al., 2023; Sun et al., 2025; Zhuang et al., 2025; Weller et al., 2025) to generate reordered document lists or coarse-grained binary relevance scores. However, they typically require larger LLMs (>7B parameters) to achieve promising results, posing challenges for real-world applications due to computational constraints.

To address this challenge, we explore using small language models (SLMs) for document reranking. Our investigation begins with a quantitative analysis. It suggests two key limitations of SLMs for document reranking: 1) SLMs are constrained by narrow representation spaces, impairing their capabilities for document reranking effectively; 2) SLMs struggle to understand task prompts and generate proper coarse-grained binary relevance scores (0: irrelevant, 1: relevant) without proper fine-tuning.

To address these limitations of SLMs, we propose a novel two-stage approach, ProRank, for SLMs document reranking. In the first stage, we employ reinforcement learning, specifically, GRPO (Group Relative Policy Optimization) (Shao et al., 2024), to teach SLMs to understand the task prompt and produce properly formatted responses, i.e., coarse-grained binary relevance scores. The GRPO learning allows ProRank to incorporate various rewards into the output format and relevance accuracy, effectively teaching SLMs to generate binary relevance scores while maximizing relevance accuracy. However, these coarse-grained binary relevance scores are insufficient for high-quality document reranking, as they merely categorize documents as relevant ("1") or irrelevant ("0") without distinguishing relevance levels among documents

[*] Work done during an internship at Mixedbread
[†] Equal Contribution
[‡] https://github.com/mixedbread-ai/mxbai-rerank
[§] https://hf.co/mixedbread-ai/mxbai-rerank-large-v2

Findings of the Association for Computational Linguistics: ACL 2026, pages 1026–1037
July 2-7, 2026

in the same category. Thus, we introduce the second stage – fine-grained score learning. We employ an efficient method to generate fine-grained relevance scores by computing the relative scores between relevant ("1") and irrelevant ("0") token logit values from the model's last token logit outputs. It maintains computational efficiency by avoiding the introduction of additional layers while providing fine-grained scoring capabilities. In our further investigation, it also helps improve SLMs' representation expressiveness for the document reranking task. We adopt the Cross-Encoder paradigm (Nogueira and Cho, 2019; Shakir et al., 2024) to efficiently train ProRank, allowing for more effective and high-quality document reranking.

For a comprehensive evaluation, we extensively experiment on three benchmarks in various languages and applications: the widely used English BEIR benchmark (Thakur et al., 2021), Chinese MTEB benchmark (CMTEB, 2025), and Code Retrieval datasets (Li et al., 2024). Extensive results demonstrate that ProRank delivers high-quality document reranking performance. Notably, our 0.5B ProRank outperforms even the 32B fine-tuned LLM model on the English BEIR benchmark. Through detailed empirical analysis, we show that ProRank effectively addresses the two key limitations of SLMs observed in the quantitative analysis.

In summary, our contributions are as follows:

- Our quantitative analysis reveals two limitations of SLMs for the document reranking: 1) they exhibit a narrow representation space, limiting their expressive capabilities; 2) they struggle with understanding task prompts.
- We propose a novel two-stage approach, ProRank, to effectively rerank documents with interpretable relevance scores, combining reinforcement learning for coarse-grained scoring with a fine-grained scoring.
- Extensive evaluations demonstrate that ProRank achieves superior document reranking performance, with 0.5B ProRank outperforming strong baselines on various benchmarks.

2 Preliminary Quantitative Analysis

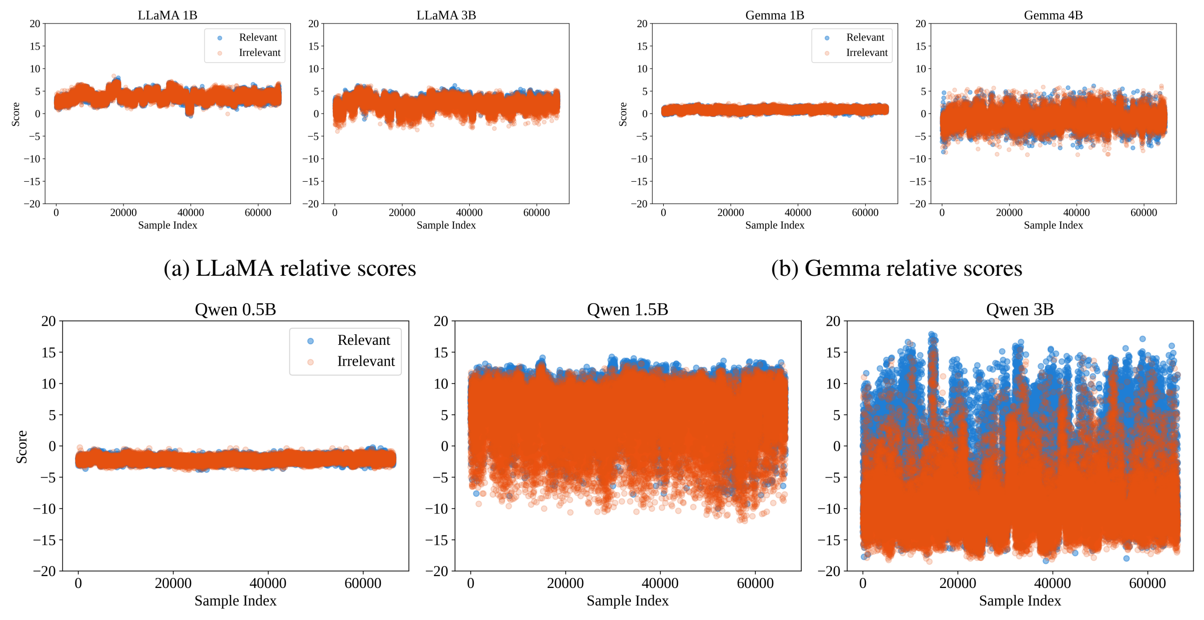
Before introducing the proposed model – ProRank, we conduct a preliminary quantitative analysis to evaluate how small language models (SLMs) perform on document reranking through zero-shot prompting. We evaluate various popular SLMs using a consistent prompt (as follows) on the TREC-COVID test set (66336 samples) from the BEIR benchmark (Thakur et al., 2021).

> **Prompt**: query: *{query_placeholder}*
> document: *{document_placeholder}*
> You are a search relevance expert who evaluates how well documents match search queries. For each query-document pair, carefully analyze the semantic relationship between them, then provide your binary relevance judgment (0 for not relevant, 1 for relevant). Relevance:

We visualize SLM's relative scores $\Delta = \mathrm{TokenLogit}(1) - \mathrm{TokenLogit}(0)$ in Figure 1, where $\mathrm{TokenLogit}(1)$ and $\mathrm{TokenLogit}(0)$ stand for the logit values corresponding to position of relevant ("1") and irrelevant ("0") tokens in the model's last token logits, respectively.

We observe two interesting phenomena: 1) **SLMs under 1B parameters exhibit a narrow representation space.** This narrow representation space could impair their effectiveness in document reranking tasks. 2) **SLMs demonstrate poor discriminative ability between "relevant" and "irrelevant"**, evidenced by many relevant document points are below irrelevant points. For phenomenon 1, it is related to the scale of models; larger models usually have more powerful representation capabilities. For phenomenon 2, it is probably a common issue for SLMs to understand the task prompt without a prompt warmup stage. To verify this, we further investigate it. Specifically, we instruct these SLMs to generate binary relevance scores for given documents and queries. We measure both the format success rate (the ability to correctly generate binary relevance scores "0" or "1") and accuracy (the correctness of relevance judgments compared to ground truth), as shown in Figure 2.

The results suggest that SLMs indeed struggle with understanding the ranking task prompt in zero-shot settings. Models like LLaMA completely fail at understanding the task, showing nearly 0% in both format success rate and accuracy. Even more powerful models like Qwen show inconsistent performance, with format success rates varying unpredictably with model size. Here, we also visualize the performance of the proposed model, ProRank, for a direct comparison. We will elaborate on it in the Section 4.5.



(a) LLaMA relative scores

(b) Gemma relative scores

(c) Qwen relative scores

Figure 1: Visualization of relative scores between relevant (“1”) and irrelevant (“0”) tokens’ logits on the TREC-COVID test set. For LLaMA, Gemma, and Qwen, we visualize the small language models (with size < 7B).

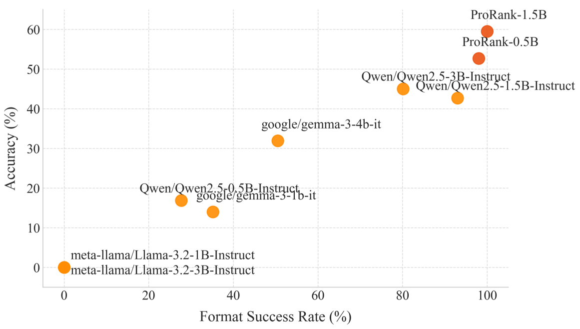


Figure 2: The accuracy (y-axis) and format success rate (x-axis) of SLMs in generating binary relevance scores on the TRECCOVID test set. The light orange colors indicate baselines’ results. The dark orange colors denote the proposed ProRank’s results.

3 Methodology

This section presents the proposed model, ProRank, with its generic framework illustrated in Figure 3. We organize as follows: Section 3.1 introduces the first stage – reinforcement learning prompt warmup, followed by Section 3.2, which details the second stage – fine-grained score learning.

3.1 Reinforcement Learning Prompt Warmup

The quantitative analysis reveals that SLMs face challenges in understanding task prompts and producing correctly binary relevance tokens without fine-tuning. Specifically, we employ GRPO (Shao et al., 2024) because it has been proven effective in optimizing multiple rewards, enabling SLMs to better learn task prompts and generate well-formatted output. The learning objective of GRPO is detailed in the appendix Section A. We fine-tune ProRank using the prompt described in Section 2. We optimize dual rewards to evaluate both format and relevance accuracy. For the format reward, we define the reward function as follows:

$$r_1(o|prompt) = \begin{cases} 1 & \text{if } o \text{ is binary token} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

We reward the model if it generates a response that follows the required binary format; otherwise, no reward will be given. For relevance accuracy, we use the accuracy of the model’s prediction as the reward function:

$$r_2(o|prompt) = \text{accuracy}(o, y) \quad (2)$$

where y is the ground truth relevance label. These dual rewards provide effective training signals throughout the reinforcement learning process, allowing SLMs to generate correctly formatted outputs while maximizing relevance accuracy for the document reranking.

3.2 Fine-grained Score Learning

While the reinforcement learning prompt warmup stage helps SLMs understand the task prompt well

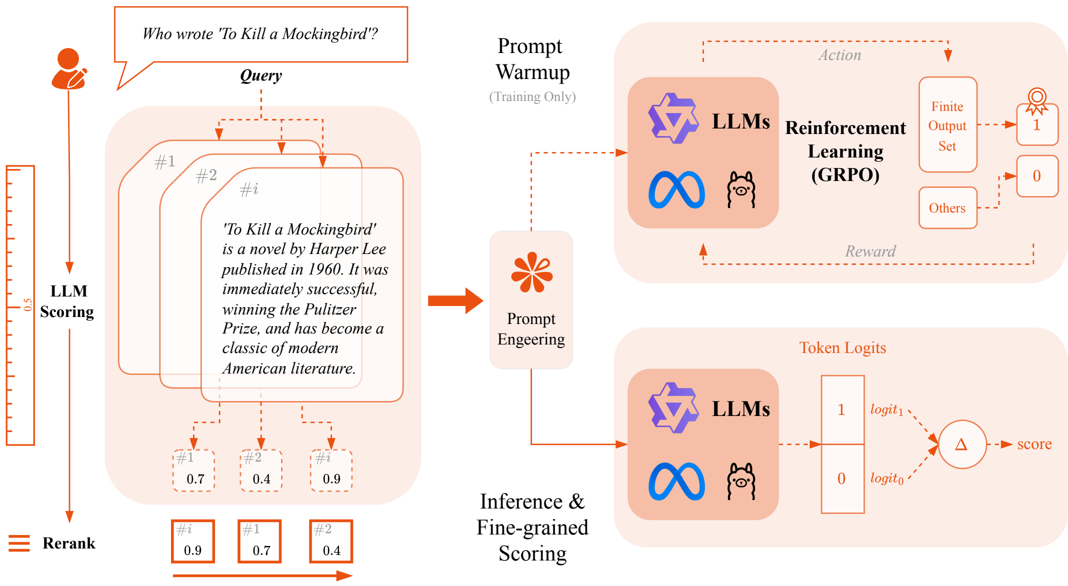


Figure 3: The generic framework of the proposed model ProRank. There are two stages: 1) Prompt Warmup (training only) steers LLM to produce coarse-grained binary relevance tokens ("1" and "0") through reinforcement learning; 2) Fine-grained scoring stage computes relative logit scores to produce fine-grained scores for ranking.

and produce correctly binary coarse-grained scores, i.e., "1" and "0". Its representation space is still narrow and lacks the necessary granularity for effective document reranking, making it difficult to rank documents with the same relevance score.

To address this limitation, we introduce a fine-grained score learning stage to enhance document reranking performance without introducing new layers. Specifically, ProRank computes a fine-grained relevance score by extracting and comparing logit values corresponding to binary tokens "0" and "1" from the last token's logits, as follows:

$$\begin{aligned} \Delta &= \mathrm{TokenLogit}(1) - \mathrm{TokenLogit}(0) \\ \mathrm{TokenLogit}(t) &= \mathbf{O}[\mathrm{token\_id}(t)] \\ \mathbf{O} &= \mathrm{LLM}_{last\_pool}(prompt) \in \mathbb{R}^V, \end{aligned} \tag{3}$$

where $\mathrm{token\_id}(t)$ maps an input token t to its index in the model's vocabulary, $prompt$ represents the input prompt, $last\_pool$ denotes the last token pooling, $\mathbf{O}$ means the logit outputs at the last token position, and V stands for the vocabulary size. We extract logit outputs from the last token position since it encapsulates the input's complete semantics. This is because in the auto-regressive architecture of LLMs, the last token can attend to all previous tokens in the attention calculation. Additionally, the reinforcement learning prompt warmup stage ensures the model generates meaningful binary tokens with the last token logits. By doing so, ProRank leverages the model's learned token logits to generate fine-grained relevance scores, without requiring additional parameters or architectural changes, making it both efficient and effective.

We train ProRank by minimizing the binary cross-entropy loss between the predicted fine-grained scores ($\hat{y}_i$) and the ground truth relevance labels (y_i), as follows:

$$\mathcal{L}_{\mathrm{BCE}} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\right]. \tag{4}$$

4 Experiment

4.1 Experiment Setup

Datasets We train ProRank using the bge-m3 dataset, following the popular reranking model *BAAI/bge-reranker-v2-m3* (Chen et al., 2024). This dataset is specifically curated for multilingual embedding and reranking tasks. For a comprehensive evaluation, we conduct experiments across multiple benchmarks spanning diverse languages and domains: (1) The BEIR benchmark (Thakur et al., 2021) for English reranking, where we utilize 14 datasets (details in Section B). We further enhance our evaluation with additional English datasets, including news, robust04, and signal; (2) Chinese retrieval datasets from the C-MTEB benchmark (CMTEB, 2025) to assess Chinese-specific rerank-

Model	Params	SF	NFC	TC	AA	FQA	QRA	DBP	TCH	SD	FVR	CFV	HQA	MM	NQ	Avg.
								First-stage Retriever								
BM25	–	67.89	33.75	59.47	29.99	23.61	78.86	31.80	44.22	14.91	65.13	16.51	63.30	22.84	30.55	41.63
								BERT-variant Reranker								
mxbai	0.4B	74.83	39.03	85.33	12.06	40.46	73.90	44.66	35.56	18.89	78.92	23.92	70.85	36.31	55.69	49.32
bge-m3	0.6B	74.5	35.89	77.74	51.98	39.50	89.02	44.05	35.14	14.91	83.45	29.19	79.5	41.73	58.56	53.94
								LLM-based Reranker								
MonoT5†	3B	76.10	37.80	79.60	42.50	46.50	–	44.50	30.70	19.30	–	25.40	–	–	–	–
RankLlama†	7B	71.10	27.00	80.20	54.40	42.10	–	43.70	41.40	16.60	–	23.20	–	–	–	–
RankLlama†	13B	72.70	28.10	80.80	49.30	44.10	–	44.90	39.20	18.10	–	24.50	–	–	–	–
bge-gemma	2.5B	78.11	38.35	78.13	54.6	41.03	89.23	44.27	32.93	19.66	84.39	31.55	80.21	41.83	60.95	55.37
								Proprietary Reranker								
cohere	–	76.58	34.65	79.5	59.90	43.40	87.12	45.12	37.20	18.70	86.20	30.10	77.30	40.60	59.09	55.39
voyage	–	76.25	35.50	79.9	61.5	46.7	87.6	43.5	28.44	18.2	85.8	21.4	78.02	41.57	59.12	54.54
								Coarse-grained Reranker								
ProRank	0.5B	76.16	37.78	77.21	59.21	40.51	87.51	42.93	31.35	17.60	88.77	31.98	78.51	37.26	56.89	54.55
ProRank	1.5B	79.76	35.90	77.70	67.89	42.25	88.95	40.97	28.00	17.66	89.87	36.40	79.23	35.62	59.95	55.73
								Fine-grained Reranker								
ProRank	0.5B	77.15	37.10	79.06	57.82	41.09	88.61	44.66	34.58	17.02	88.55	33.02	79.14	41.25	58.89	55.57
ProRank	1.5B	80.87	37.06	80.01	67.97	44.41	89.26	45.63	32.06	17.83	89.86	37.12	79.66	41.75	61.43	57.49

Table 1: Results on BEIR benchmark, with NDCG@10 as the main metric. The orange color indicates the best values. † indicates results are retrieved from (Zhuang et al., 2025). Our results are the average of five runs. Our 1.5B ProRank achieves the best result with significant performance gains compared to baselines on average ($p < 1\%$).

ing capabilities, specifically Covid and DuReader; and (3) the COSQA dataset from the COIR benchmark (Li et al., 2024) to evaluate code reranking performance. This diverse evaluation strategy enables us to thoroughly assess our model's generalizability across languages and domains, providing insights into its robustness and versatility in real-world applications.

Baselines To establish a comprehensive evaluation, we compare our model against popular state-of-the-art reranking models across different parameter scales. Our baselines include open-source models of varying sizes: *mixedbread-ai/mxbai-rerank-large-v1* (Shakir et al., 2024) (mxbai), a BERT-scale model with 435M (0.4B) parameters known for its strong generalization capabilities; *BAAI/bge-reranker-v2-m3* (Chen et al., 2024) (bge-m3), a cutting-edge reranker with 568M (0.6B) parameters that represents the current state-of-the-art at the BERT scale; and *BAAI/bge-reranker-v2-gemma* (Chen et al., 2024) (bge-gemma), a larger 2.5B parameter SLM-based reranker that demonstrates the capabilities of models with significantly more parameters. Additionally, we benchmark against two famous proprietary commercial reranking mdoels, including *cohere-rerank-3.5* (Cohere, 2024) (cohere) and *voyage-rerank-2* (Voyage, 2024) (voyage), to position our approach within the broader landscape of available reranking solutions.

Evaluation Metric We report Normalized Discounted Cumulative Gain at rank 10 (NDCG@10), which measures the quality of ranking by considering both relevance and position, with higher scores indicating better ranking quality.

Implementation Details We use the popular Qwen (Yang et al., 2024) as the backbone model and support two parameter scales: 0.5B and 1.5B. For efficient tuning, we employ LoRA (Hu et al., 2022) with hyperparameters lora_r=32, lora_alpha=32, and lora_dropout=0.1 after a mild hyperparameter sweep. The training process uses AdamW optimizer (Loshchilov and Hutter, 2019) with an initial learning rate of $1e-4$ for both stages. Following the common evaluation practice, we use BM25 as the first-stage retriever to retrieve initial document candidates. Following common practice, we retrieve the top 100 documents for each query. All reranking models are then evaluated on the same retrieved documents for a fair comparison.

4.2 Main Results

Results on BEIR Benchmark We first present our main results on the BEIR benchmark in Table 1. When comparing LLM-based models with

Model	Params	news EN	robust04 EN	signal EN	Covid CN	DuReader CN	COSQA Code	Avg.
				First-stage Retriever				
BM25	–	39.52	40.70	33.05	76.10	53.39	21.79	44.09
mxbai	0.4B	39.52	56.38	31.98	78.60	66.46	30.72	50.61
bge-m3	0.6B	43.21	50.24	30.66	85.94	77.72	24.86	52.11
bge-gemma	2.5B	47.22	51.94	32.84	78.55	78.44	31.51	53.42
				Coarse-grained Reranker				
ProRank	0.5B	47.54	51.45	33.85	85.44	76.52	30.31	54.19
ProRank	1.5B	47.28	53.36	30.96	85.55	75.46	29.35	53.66
				Fine-grained Reranker				
ProRank	0.5B	46.57	52.25	34.12	89.37	78.03	31.73	55.35
ProRank	1.5B	49.06	54.32	31.85	89.78	78.54	32.05	55.93

Table 2: Results on other English, Chinese, and code retrieval datasets, with NDCG@10 as the primary metric. The orange color indicates the best values across all models.

BERT-variant models, we observe that LLM-based models generally achieve higher average scores across the benchmark. This performance advantage can be attributed to LLMs' larger parameter sizes and enhanced generalization capabilities, highlighting their effectiveness for document reranking. Interestingly, we note that the BERT-variant model *mixedbread-ai/mxbai-rerank-large-v1* still outperforms the state-of-the-art bge-gemma reranking model on five specific datasets. This finding suggests considerable room for improvement in existing LLM-based reranking approaches. Our 1.5B ProRank model with fine- and coarse-grained scoring demonstrates superior performance compared to baseline approaches, while even our smaller 0.5B variant remains competitive. These results strongly validate the effectiveness of our proposed two-stage training methodology. We also find that our ProRank with fine-grained score consistently outperforms the coarse-grained score, suggesting the importance of fine-grained scoring for effective document reranking.

Results on Additional Languages and Domains Table 2 presents the results on diverse datasets of English, Chinese, and code retrieval. Our proposed ProRank achieves superior performance compared to all baselines on nearly all datasets, with the only exception being the robust04 dataset, where *mxbai* shows slightly better results. The performance improvements are particularly notable for our fine-grained score models, with the 0.5B and 1.5B ProRank delivering average gains of 1.93% and 2.51%, respectively, over the powerful baseline–*bge-gemma*. These findings further underscore the generalizability and effectiveness of ProRank in handling diverse languages and domains.

4.3 Ablation Study

To better understand the contribution of each component in ProRank, we conduct a comprehensive ablation study. First, we examine the impact of the fine-grained score learning stage. As shown in Table 1, the fine-grained score learning significantly enhances reranking quality, confirming its critical role in our two-stage methodology.

To verify the importance of reinforcement learning prompt warmup, we train a 0.5B Qwen model without this initial stage. It achieves a 2.04% improvement when using reinforcement prompt warmup compared to without it, demonstrating that reinforcement learning prompt warmup is important for achieving better reranking performance.

We also compare different fine-tuning strategies, specifically evaluating supervised fine-tuning (SFT) and the popular reinforcement learning algorithm–GRPO. The detailed comparison is in Appendix Section C. The results indicate that using reinforcement learning for the first-stage prompt warmup achieves better performance than SFT, benefiting task prompt understanding.

4.4 Discussion on Representation Capabilities

In the quantitative analysis, we observed that SLMs exhibit a narrow representation space without fine-tuning, resulting in limited representation expressiveness. Here, we visualize ProRank's relative scores in Figure 4 to investigate this phenomenon.

For the 0.5B model, we observe a progressive widening of the representation space across training stages. It indicates that ProRank gradually learns to better distinguish between relevant and irrelevant documents, demonstrating improved representation expressiveness for the document reranking task.

For the 1.5B model, the pattern differs. The base Qwen 1.5B model already exhibits a relatively wide representation space, suggesting that the larger model possesses sufficient inherent expressiveness for the document reranking task. Consequently, the representation space does not undergo the same dramatic widening observed in the 0.5B ProRank across training stages. Nevertheless, the relative scores of relevant document points consistently shift upward and concentrate above irrelevant document points, suggesting the two-stage training process successfully teaches it to understand the task prompt, ultimately achieving high-quality document reranking through proper score calibration rather than representational expansion.

This evidence suggests that the proposed ProRank can effectively address the observed limitations in quantitative analysis.

4.5 Discussion on Task Understanding

In the quantitative analysis, we found that SLMs struggle with understanding task prompts. To evaluate this, we visualize the format success rate of ProRank in Figure 2. It shows that through the stage one prompt warmup learning, ProRank effectively learns to understand task prompts. Furthermore, as shown in Figure 4, both 0.5B and 1.5B ProRank demonstrate clear improvements in document ranking ability, with the relative scores of relevant document points increasing and clustering above those of irrelevant document points. This indicates that ProRank progressively develops a strong understanding of the document reranking task through the two-stage training process.

4.6 Discussion on Top-k Retrieval

Following Jacob et al. (2024), we test different top-k values (10, 100, 1,000, 5,000) on SciFact and FiQA datasets from BEIR to study how the number of retrieved documents affects reranking, as shown in Figure 5. First, all reranking models demonstrate substantial performance gains when increasing the top-k from 10 to 100 documents. It indicates that first-stage retrievers often miss highly relevant documents in the top 10; therefore, having a larger candidate pool of up to 100 is essential for effective reranking. Second, our proposed ProRank consistently outperforms all baselines across all tested top-k values on both datasets. While the initial increase in top-k is beneficial, we observe marginal effects beyond top-k=100 for both datasets. This evidence suggests that while increasing the number of candidates by expanding the top-k can improve ranking quality, it also introduces a higher volume of non-relevant documents (noise). This noise can challenge the reranking models, potentially diluting the impact of truly relevant documents, especially at very large top-k values like 5,000. Our ProRank, while generally robust, also shows sensitivity to noise at higher top-k values, highlighting an ongoing challenge in document reranking.

4.7 More Discussions

To gain more insights, we present a detailed case study in Appendix Section D, analyze the performance differences between coarse- and fine-grained scoring in Appendix Section E, compare ProRank with cutting-edge LLMs for document reranking task in Appendix Section F, and discuss the efficiency of ProRank in Appendix Section G.

5 Related Work

This work is in line with document rerankinga process that typically serves as the second stage in information retrieval pipelines. After an initial retrieval step—such as BM25 or dense retrieval (Li et al., 2025; Li and Li, 2024, 2023; Gao et al., 2021), reranking is employed to refine and improve the quality of the results. Traditionally, reranking has been performed using BERT-based models. Nogueira and Cho (2019) pioneered this approach with BERT for passage reranking, establishing a framework where query-document pairs are encoded together to generate relevance scores. These models, often referred to as Cross-Encoders, have become the standard for traditional reranking systems. Recent advancements, such as *mixedbread-ai/mxbai-rerank* (Shakir et al., 2024) and *bge-reranker* (Chen et al., 2024), optimize the Cross-Encoder architecture for improved generalization across diverse retrieval tasks.

The emergence of large language models (LLMs) has created new paradigms for reranking. Zero-shot approaches leverage LLMs' inherent capabilities without task-specific training (Zhuang et al., 2023; Ma et al., 2023). Sun et al. (2023) demonstrated that LLMs like ChatGPT can effectively rerank documents through carefully designed prompting engineering. Zhuang et al. (2024) further advanced this direction with setwise ranking

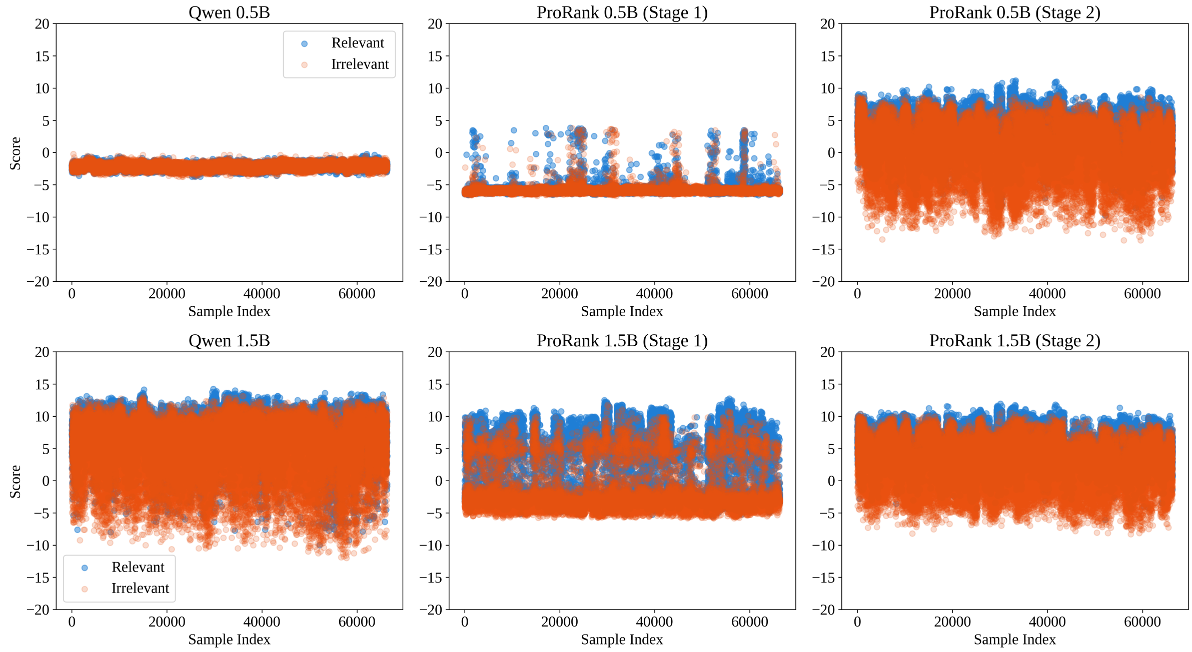
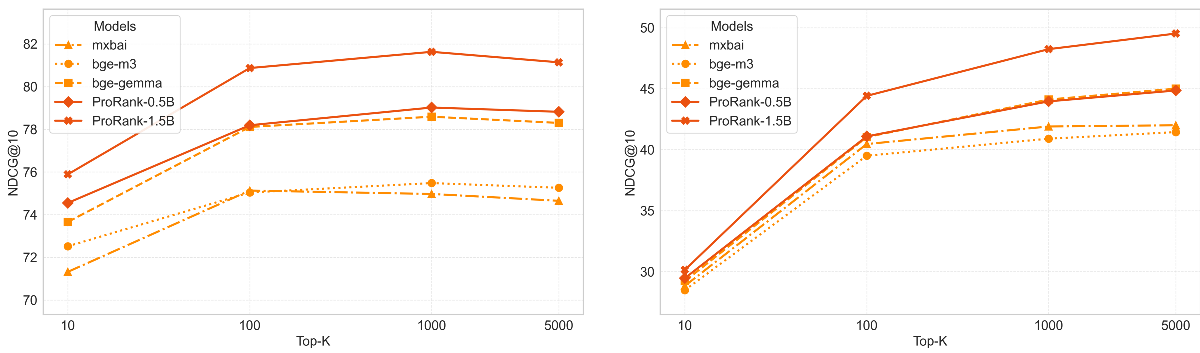


Figure 4: Visualization of relative scores between relevant ("1") and irrelevant ("0") tokens' logits on the TREC-COVID test set for Qwen and ProRank.



(a) Top-k Retrieval on SciFact

(b) Top-k Retrieval on FiQA

Figure 5: Performance comparison of our fine-grained reranker across different numbers of retrieved documents (top-k) on the SciFact and FiQA dataset.

approaches that consider multiple documents simultaneously. To further improve the performance of LLMs for reranking, recent work has focused on fine-tuning LLMs for reranking. *Rank1* (Weller et al., 2025) introduced techniques to optimize compute efficiency during inference time. Zhuang et al. (2025) proposed *Rank-R1*, which enhances LLM reasoning capabilities for document reranking through reinforcement learning. Commercial models like *cohere-rerank-3.5* (Cohere, 2024) and *voyage-rerank-2* (Voyage, 2024) offer powerful document reranking capabilities and strong generalization ability.

Despite these advancements, current reranking approaches still have limitations. Zero-shot methods require larger LLMs to function effectively, but these models are challenging to interpret and computationally expensive to utilize in practice (Zhu et al., 2023). Our proposed ProRank differs from existing work by enabling small language models (SLMs) to perform high-quality reranking while producing interpretable, fine-grained relevance scores, addressing significant limitations in current LLM-based reranking systems.

6 Conclusion

In this paper, we present ProRank, a novel two-stage approach for document reranking using Small Language Models (SLMs). Through a preliminary quantitative analysis, we identify two key limitations of SLMs in document reranking: inadequate task prompt understanding and constrained representation space. To address the first limitation, ProRank uses reinforcement learning for prompt warmup learning. Additionally, ProRank employs a fine-grained scoring mechanism to expand the representation space, enhancing representation expressiveness and delivering superior document reranking performance. Comprehensive experiments on various domains and languages have verified the effectiveness and generalizability of ProRank. By enabling SLMs to achieve competitive performance, ProRank makes high-quality reranking more accessible for resource-constrained environments.

Limitations

A potential limitation observed in our experiments (Section 4.6) is the challenge of handling noise in scenarios with very large top-k (e.g., $k = 5,000$) retrieval. To address this challenge, we recognize the need for enhanced semantic understanding capabilities that can better distinguish relevant results from noise. For this limitation, our future work will pursue several promising directions: (1) developing more noise-robust reranking architectures, (2) investigating adaptive top-k selection mechanisms that dynamically adjust based on query characteristics, and (3) extending ProRank's applicability to smaller, more efficient models and novel application domains.

Acknowledgements

This work was supported by Mixedbread AI.

Xianming Li and Jing Li are partially supported by the grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. PolyU/25200821, Project No. T41-517/25-N), the Innovation and Technology Fund (Project No. PRP/047/22FX).

Here, we sincerely thank the reviewers and ACs for their valuable input, which has greatly improved our work.

References

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.

CMTEB. 2025. Chinese massive text embedding benchmark.

Cohere. 2024. Introducing rerank 3.5: Precise ai search.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *ArXiv*, abs/2104.08821.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Mathew Jacob, Erik Lindgren, Matei Zaharia, Michael Carbin, Omar Khattab, and Andrew Drozdov. 2024. Drowning in documents: consequences of scaling reranker inference. *arXiv preprint arXiv:2411.11767*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Xiangyang Li, Kuicai Dong, Yi Quan Lee, Wei Xia, Hao Zhang, Xinyi Dai, Yasheng Wang, and Ruiming Tang. 2024. Coir: A comprehensive benchmark for code information retrieval models. *arXiv preprint arXiv:2407.02883*.

Xianming Li and Jing Li. 2023. Bellm: Backward dependency enhanced large language model for sentence embeddings. In *North American Chapter of the Association for Computational Linguistics*.

Xianming Li and Jing Li. 2024. Aoe: Angle-optimized embeddings for semantic textual similarity. In *Annual Meeting of the Association for Computational Linguistics*.

Xianming Li, Zongxi Li, Jing Li, Haoran Xie, and Qing Li. 2025. Ese: Espresso sentence embeddings. In *International Conference on Learning Representations*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. *7th International Conference on Learning Representations, ICLR2019*.

Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. 2024. Fine-tuning llama for multi-stage text retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2421–2425.

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

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*.

A Shakir, D Koenig, J Lipp, and S Lee. 2024. Boost your search with the crispy mixedbread rerank models.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Shuoqi Sun, Shengyao Zhuang, Shuai Wang, and Guido Zuccon. 2025. An investigation of prompt variations for zero-shot llm-based rankers. *Advances in Information Retrieval - 47th European Conference on Information Retrieval, ECIR 2025*, 15573:185–201.

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.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*.

Voyage. 2024. rerank-2 and rerank-2-lite: the next generation of voyage multilingual rerankers.

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

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 40 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Haonan Chen, Zheng Liu, Zhicheng Dou, and Ji-Rong Wen. 2023. Large language models for information retrieval: A survey. *arXiv preprint arXiv:2308.07107*.

Shengyao Zhuang, Bing Liu, Bevan Koopman, and Guido Zuccon. 2023. Open-source large language models are strong zero-shot query likelihood models for document ranking. *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8807–8817.

Shengyao Zhuang, Xueguang Ma, Bevan Koopman, Jimmy Lin, and Guido Zuccon. 2025. Rank-r1: Enhancing reasoning in llm-based document rerankers via reinforcement learning. *arXiv preprint arXiv:2503.06034*.

Shengyao Zhuang, Honglei Zhuang, Bevan Koopman, and Guido Zuccon. 2024. A setwise approach for effective and highly efficient zero-shot ranking with large language models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 38–47.

A GRPO Learning

We use reinforcement learning in the first stage of ProRank to teach small language models (SLMs) to effectively understand document ranking task prompts and produce proper signal tokens for document reranking. Specifically, we adopt GRPO (Group Relative Policy Optimization) (Shao et al., 2024), which is a popular reinforcement learning algorithm that has been proven effective for optimizing multiple rewards simultaneously (Shao et al., 2024). The GRPO learning objective is as follows:

$$
\begin{aligned}
J_{\text{GRPO}}(\theta) &= \mathbb{E}\left[q \sim P(Q), \{o_i\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}(O|q)\right] \\
&\frac{1}{G}\sum_{i=1}^{G}\Bigg(\min\Big(\frac{\pi_\theta(o_i|q)}{\pi_{\theta_{\text{old}}}(o_i|q)}A_i, \\
&\text{clip}\left(\frac{\pi_\theta(o_i|q)}{\pi_{\theta_{\text{old}}}(o_i|q)}, 1-\varepsilon, 1+\varepsilon\right)A_i\Big) \\
&-\beta\,\mathbb{D}_{\text{KL}}(\pi_\theta \,\|\, \pi_{\text{ref}})\Bigg), \\
&\mathbb{D}_{\text{KL}}\left(\pi_\theta \,\|\, \pi_{\text{ref}}\right) = \frac{\pi_{\text{ref}}(o_i|q)}{\pi_\theta(o_i|q)} \\
&- \log\frac{\pi_{\text{ref}}(o_i|q)}{\pi_\theta(o_i|q)} - 1,
\end{aligned} \tag{5}
$$

where ε and β are hyperparameters, and A_i represents the advantage, calculated using rewards $\{r_1, r_2, \ldots, r_G\}$ corresponding to outputs within each group:

$$
A_i = \frac{r_i - \text{mean}(\{r_1, r_2, \ldots, r_G\})}{\text{std}(\{r_1, r_2, \ldots, r_G\})}. \tag{6}
$$

B BEIR Benchmark Detail

In this paper, we adopt 14 datasets from the BEIR benchmark (Thakur et al., 2021): SciFact (SF), NFCorpus (NFC), TRECCOVID (TC), ArguAna (AA), FiQA2018 (FQA), Quora (QRA), DBPedia (DBP), Touche2020 (TCH), SciDOCS (SD), Fever (FVR), Climate-Fever (CFV), HotpotQA (HQA), MSMARCO-dev (MM), and Natural Questions (NQ).

C SFT *vs* GRPO

Here, we compare 0.5B ProRank with different prompt warmup training: supervised fine-tuning

Model	Params	SF	NFC	TC	AA	FQA	QRA	DBP	TCH	SD	FVR	CFV	HQA	MM	NQ	Avg.
ProRank (fine-grained)																
+ SFT	0.5B	76.9	36.9	77.51	62.51	41.16	80.75	43.53	36.15	15.47	81.32	27.49	78.92	40.92	58.03	54.11
+ GRPO	0.5B	77.15	37.10	79.06	57.82	41.09	88.61	44.66	34.58	17.02	88.55	33.02	79.14	41.25	58.89	55.57

Table 3: Results of ProRank with SFT and GRPO prompt warmup on BEIR benchmark, with NDCG@10 as the main metric.).

Document	**HM**	**MB**	**M3**	**GEM**	**CH**	**VY**	**PR$^C_{0.5B}$**	**PR$^C_{1.5B}$**	**PR$^F_{0.5B}$**	**PR$^F_{1.5B}$**
"To Kill a Mockingbird' is a novel by **Harper Lee** published in 1960.'	0	0	0	0	0	0	1	1	0	0
Harper Lee, an American novelist, was born in 1926 in Monroeville, Alabama.	1	1	2	1	1	1	0	0	1	1
The 'Harry Potter' series, consisting of seven fantasy novels, was written by British author J.K. Rowling.	2	3	1	3	2	3	2	2	2	2
What does a Mockingbird eat?	3	2	3	2	3	2	3	3	3	3

Table 4: Case study of different model rankings for query *"Who wrote 'To Kill a Mockingbird'?"* HM: Human; MB: *mxbai-rerank-large-v1*; M3: *bge-rerank-v2-m3*; GEM: *bge-rerank-v2-gemma*; CH: *cohere-rerank-3.5*; VY: *voyage-rerank-2*; PR$^C_{0.5B}$: coarse-grained *ProRank-0.5B*; PR$^C_{1.5B}$: coarse-grained *ProRank-1.5B*; PR$^F_{0.5B}$: fine-grained *ProRank-0.5B*; PR$^F_{1.5B}$: fine-grained *ProRank-1.5B*.

(SFT) and popular GRPO. The results are listed in Table 3. We can find that ProRank with GRPO prompt warmup performs better than the SFT one. It might be attributed to the multiple rewards optimization strategy, making the model more relevant and accurate in the first stage of training, thereby improving the second stage's performance. This also demonstrates the superiority of reinforcement learning in the first stage of training.

D Case Study

Here, we conduct a case study, as shown in Table 4, to intuitively compare the performance of different models on document reranking.

For baselines, we find that *bge-m3* performs poorly on the second document's ranking. Although the second document does not explicitly mention "Mockingbird", it should be ranked higher than the third and fourth documents since it directly identifies **Harper Lee** as the author. This might be attributed to limited commonsense understanding of *bge-m3*. The *bge-gemma* and proprietary model *voyage-rerank-2* rank the fourth document (containing "Mockingbird") higher than other models. This could indicate a greater reliance on lexical overlap rather than fine-grained semantic matches

For the proposed model, while the coarse-grained ProRank assigns a score of 1 (relevant) to both the first and second documents, it is impossible to differentiate their relative relevance. The fine-grained version of ProRank addresses this limitation by providing more nuanced relevance scores. Notably, the proposed ProRank with fine-grained scores produces rankings that closely align with human judgment, validating its effectiveness.

E Discussion on Score Granularity

Here, we deep into score granularity's impact on reranking performance. Table 1 demonstrates that ProRank with the fine-grained scoring version consistently outperforms the coarse-grained one. For 0.5B ProRank, the fine-grained scoring version achieves a 1.02% absolute improvement in average NDCG@10 (55.57% *vs* 54.55%). This improvement becomes more significant with 1.5B ProRank, where the fine-grained scoring version delivers a 1.76% improvement (57.49% *vs* 55.73%). Table 2 further confirms these findings across languages and domains. The average improvement from the fine-grained scoring version is 1.16% for 0.5B ProRank and 2.27% for 1.5B ProRank. These results demonstrate that fine-grained scoring effectively improves document reranking quality. It may be because it captures subtle relevance differences and provides more informative training signals, aligning better with human judgment patterns.

F Comparison with Cutting-edge LLMs

We further evaluate ProRank by comparing it with two powerful LLMs: Gemini and GPT. The results, presented in Table 5, reveal a significant advantage for ProRank in pointwise document reranking. For instance, on SciFact, ProRank 0.5B surpasses Gemini and GPT by 9.18% and 5.77%, respectively; the larger ProRank 1.5B achieves even greater improvements of 12.18% and 8.77%. Notably, ProRank attains these results in mere minutes, compared to the hours required by the LLMs. This combination of superior effectiveness and drastically lower computational cost not only highlights ProRank's superiority for reranking but also confirms its strong practical applicability.

Model	SciFact	Time (s)
Gemini-2.5-flash	67.97	3:33:59
GPT5.1	71.38	5:55:18
ProRank-0.5B	77.15	00:03:01
ProRank-1.5B	80.15	00:06:17

Table 5: Results of Gemini, GPT, and ProRank on the SciFact dataset. NDCG@10 serves as the metric.

Model	Params	Load (s)	Avg. (ms)
bge-v2-gemma	2B	1.26	64.0
rankllama-7b	7B	4.88	49.7
rankllama-13b	13B	6.77	81.0
ProRank-0.5B	0.5B	1.29	27.2
ProRank-1.5B	1.5B	1.55	32.7

Table 6: Latency of different scales of LLM rerankers. rankllama is from (Ma et al., 2024). Avg. means average latency.

G Discussion on Efficiency

Following the scaling law (Kaplan et al., 2020), smaller models are generally more efficient. We quantitatively compare the efficiency of different-scale LLM rerankers, as detailed in Table 6. The results confirm that our smaller-scale ProRank is significantly more efficient than baseline rerankers using larger-scale LLMs.