TFRank: Think-Free Reasoning Enables Practical Pointwise LLM Ranking

Yongqi Fan[1,2*†], **Xiaoyang Chen**[3,4,2*†], **Dezhi Ye**[2], **Jie Liu**[2], **Haijin Liang**[2], **Jin Ma**[2], **Ben He**[3,4], **Yingfei Sun**[3], **Tong Ruan**[1‡]

[1] East China University of Science and Technology, Shanghai, China
[2] Tencent
[3] University of Chinese Academy of Sciences
[4] Chinese Information Processing Laboratory, Institute of Software, Chinese Academy of Sciences
johnnyfans@mail.ecust.edu.cn, chenxiaoyang19@mails.ucas.ac.cn, ruantong@ecust.edu.cn

Abstract

Reasoning-intensive ranking models built on Large Language Models (LLMs) have made notable progress. However, existing approaches often rely on large-scale LLMs and explicit Chain-of-Thought (CoT) reasoning, resulting in high computational cost and latency that limit real-world use. To address this, we propose **TFRank**, an efficient pointwise reasoning ranker based on small-scale LLMs. To improve ranking performance, TFRank effectively integrates CoT data, fine-grained score supervision, and multi-task training. Furthermore, it achieves an efficient "**T**hink-**F**ree" reasoning capability by employing a "think-mode switch" and pointwise format constraints. Specifically, this allows the model to leverage explicit reasoning during training while delivering precise relevance scores for complex queries at inference without generating any reasoning chains. Experiments show that TFRank achieves performance comparable to models with four times more parameters on the BRIGHT benchmark and demonstrates strong competitiveness on the BEIR benchmark. Further analysis shows that TFRank achieves an effective balance between performance and efficiency, providing a practical solution for integrating advanced reasoning into real-world systems.

Code — https://github.com/JOHNNY-fans/TFRank
Extended version — https://arxiv.org/abs/2508.09539

Introduction

Driven by the advancements in Large Language Models (LLMs), Information Retrieval (IR) systems are increasingly confronted with the demand to handle complex, inferential user queries (DeepSeek-AI et al. 2025; Yang et al. 2025a; Su et al. 2025; Weller et al. 2025a). In modern applications such as Retrieval-Augmented Generation (RAG) systems and Multi-Agent frameworks (Yu et al. 2024; Chen et al. 2024b; Guo et al. 2024), the ability to perform reasoning over both queries and retrieved documents is essential for ranking models to deliver accurate and relevant information. This has led to research on *reasoning-intensive ranking*, which aims to develop ranking algorithms capable of understanding subtle relationships and accurately assessing the relevance of documents to complex information needs.

Early attempts to endow rankers with reasoning capabilities relied primarily on the power of LLMs. Methods leverage LLMs' instruction-following abilities to flexibly judge query–document relevance (Sun et al. 2023; Ma et al. 2023; Yu et al. 2024). To further enhance performance on complex queries, many approaches explicitly incorporate the Chain-of-Thought (CoT) mechanism. By generating reasoning text before making a final relevance judgment, models such as Rank1 (Weller et al. 2025b), Rank-R1 (Zhuang et al. 2025), and REARANK (Zhang et al. 2025a) have demonstrated significant performance improvements in handling queries that require sophisticated reasoning.

However, the pursuit of performance often incurs prohibitive costs, hindering the transition from research prototypes to deployable systems. A number of state-of-the-art rankers rely on computationally and financially expensive models, whether proprietary (e.g., GPT-4) or massive open-source versions (e.g., Llama-3-70B-Instruct (Dubey et al. 2024)). Moreover, the reasoning process that boosts performance is token-intensive, leading to significant inference latency. Finally, listwise or setwise approaches, adopted for cross-document comparison, exhibit limited scalability, rendering them unsuitable for low-latency, real-time ranking.

Therefore, as illustrated in Figure 1, the core challenge in designing a practical, reasoning-capable ranker is: **How can we leverage the advantages of reasoning without sacrificing efficiency and robustness in real-world scenarios?** A deployable ranker must meet two essential requirements. It should offer strong performance, including robust query-document relevance modeling, multi-document comparison, reasoning-based inference, and fine-grained scoring. It must also satisfy strict efficiency constraints, such as compact model size, short output length, stable formatting, and a scalable pointwise ranking architecture.

To address this challenge, we propose a novel training and inference pipeline for *pointwise* ranking that balances efficiency and performance with small-scale LLMs. In the training phase, we adopt a multi-task strategy that inte-

[*] Co-first authors.
[†] Work was done when interning at Tencent.
[‡] Corresponding author.

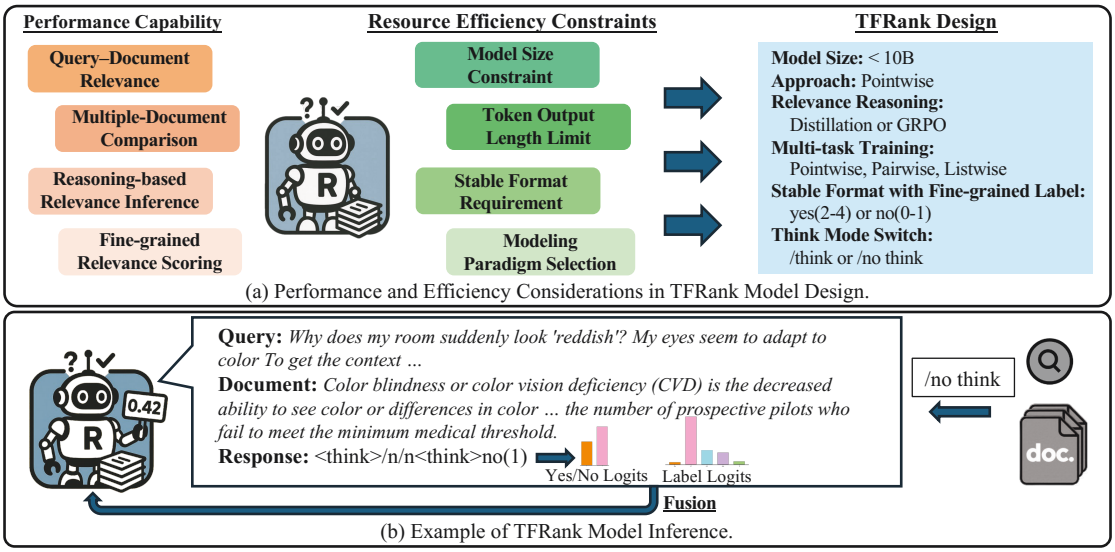

Figure 1: Overview of the TFRank framework. (a) summarizes key performance and efficiency considerations in the model design. (b) provides an example of inference with the TFRank model.

grates pointwise, pairwise, and listwise supervision with fine-grained relevance labels and CoT signals, enabling the model to learn nuanced and precise scoring. To further strengthen the model's capability, we incorporate an optional GRPO (Shao et al. 2024) training for advanced reasoning optimization. At inference, a "think-mode switch" and pointwise output constraints prompt the model to generate direct relevance scores for each query–document pair, fully bypassing explicit reasoning generation.

Experiments on the reasoning-intensive BRIGHT benchmark reveal a striking finding: our "Think-Free" inference mode, which omits explicit reasoning chains, not only dramatically improves efficiency but also surpasses the ranking performance of the full-reasoning mode. We term this phenomenon **"Think-Free" Reasoning** and name our model **TFRank** accordingly. This suggests the model internalizes its reasoning abilities via multi-task training, enabling it to produce high-quality judgments directly. The efficacy of TFRank is particularly pronounced on smaller models: for instance, our 1.7B TFRank model competes with 7B-parameter baselines on BRIGHT, models over four times its size. Moreover, on the general BEIR benchmark, TFRank maintains comparable performance to existing methods while offering substantial efficiency gains. Thus, TFRank provides a validated and efficient pathway for building a new generation of ranking systems that combine state-of-the-art reasoning capabilities with production-level feasibility.

Our contributions can be summarized as follows:

- We propose **TFRank**, a novel and efficient framework that enables small-scale LLMs to perform reasoning-intensive ranking by internalizing reasoning patterns through multi-task training.
- We identify the **"Think-Free" Reasoning** phenomenon, where suppressing explicit reasoning chains at inference leads to both superior ranking accuracy and dramatic efficiency gains.
- Our experiments demonstrate that TFRank achieves promising results on the BRIGHT benchmark, with a 1.7B model competing with 7B baselines, proving its practical viability for production systems.

Related Works

Application of LLMs in Ranking LLM-based ranking methods can be categorized into three paradigms according to how documents are processed. The **pointwise** paradigm evaluates each document independently for query relevance and outputs a score (Ma et al. 2024; Li et al. 2025; Weller et al. 2025b; Zhang et al. 2025b; Liu, Zhu, and Dou 2024), offering high parallelism and low latency, which makes it suitable for real-time applications. The **pairwise** paradigm compares document pairs to provide finer-grained supervision (Qin et al. 2024; Luo et al. 2024), but its quadratic complexity limits scalability with many candidates. **Listwise** and **setwise** methods (Sun et al. 2023; Ma et al. 2023; Zhuang et al. 2024) process multiple documents jointly, but face increased input/output length and lower efficiency due to sliding-window strategies, hindering real-time deployment. Recent studies further explore long-context LLMs to fully rank candidate passages (Liu et al. 2025c), which improves efficiency over sliding windows but inevitably increases latency. In contrast, our proposed TFRank adopts an efficient pointwise architecture while integrating strong reasoning abilities through targeted training.

Reasoning-Intensive LLM Ranking With the rapid progress of RAG and Multi-Agent systems, as well as the emergence of stronger LLMs (e.g., DeepSeek-R1 (DeepSeek-AI et al. 2025), OpenAI o1), explicit reasoning has become increasingly integrated into ranking models (Zhu et al. 2023). Existing research follows two main di-

Model		Approach	StackExchange							Coding		Theorem-based			Avg.
			Bio.	Earth.	Econ.	Psy.	Rob.	Stack.	Sus.	Leet.	Pony	AoPS	TheoQ.	TheoT.	
BM25	-	Pointwise	18.2	27.9	16.5	13.4	10.9	16.3	16.1	24.7	4.3	6.5	7.3	2.1	13.7
RankGPT-4	Zero-shot	Listwise	**33.8**	34.2	16.7	27.0	22.3	**27.7**	11.1	3.4	15.6	1.2	0.2	8.6	17.0
Deepseek-R1-rank_llm[1]	Zero-shot	Listwise	14.6	16.9	11.1	17.3	17.5	11.2	15.3	8.1	8.8	5.6	4.7	8.1	11.6
Rank1-7B	SFT	Pointwise	31.4	36.7	18.3	25.4	13.8	17.6	24.8	16.7	9.5	6.1	9.5	11.6	18.5
Rank1-14B	SFT	Pointwise	29.6	34.8	17.2	24.3	18.6	16.2	24.5	17.5	14.4	5.5	9.2	10.7	18.5
REARANK-7B	GRPO	Listwise	23.4	27.4	18.5	24.2	17.4	16.3	25.1	**27.0**	8.0	7.4	7.9	9.5	17.7
Rank-R1-3B	GRPO	Setwise	18.4	17.1	13.7	16.9	9.0	10.0	16.5	11.1	4.7	3.5	3.2	5.9	10.8
Rank-R1-7B	GRPO	Setwise	26.0	28.5	17.2	24.2	19.1	10.4	24.2	19.8	4.3	4.3	8.3	10.9	16.4
Rank-R1-14B	GRPO	Setwise	31.2	38.5	21.2	26.4	**22.6**	18.9	**27.5**	20.2	9.2	9.7	9.2	**11.9**	20.5
Llama3.2 Series															
TFRank-1B	SFT	Pointwise	23.8	25.5	8.6	15.3	7.3	11.0	8.9	9.4	9.3	4.4	6.2	2.2	11.0
TFRank-3B	SFT	Pointwise	31.3	42.2	19.1	27.5	14.7	17.9	19.9	19.9	2.8	6.7	9.2	8.0	18.3
TFRank-8B	SFT	Pointwise	31.8	41.5	19.7	**30.0**	17.2	20.9	19.2	26.0	**20.3**	**10.7**	10.3	9.6	**21.4**
Qwen2.5 Series															
TFRank-0.5B	SFT	Pointwise	16.7	23.0	8.3	14.4	13.7	8.1	14.1	9.0	15.2	3.4	2.8	1.1	10.8
TFRank-1.5B	SFT	Pointwise	22.1	23.3	15.9	20.3	11.8	13.6	14.3	17.8	17.5	6.7	9.5	6.9	15.0
TFRank-3B	SFT	Pointwise	29.5	36.2	16.8	24.3	15.1	13.8	18.7	23.1	8.6	8.9	9.3	9.1	17.8
TFRank-7B	SFT	Pointwise	31.0	41.2	20.1	26.8	19.8	18.1	22.0	23.4	16.9	7.0	10.0	10.5	20.6
Qwen3 Series															
TFRank-0.6B	SFT	Pointwise	24.8	30.0	12.0	17.5	12.9	12.1	12.7	24.4	13.1	7.1	10.3	9.8	15.6
TFRank-1.7B	SFT	Pointwise	25.2	29.7	17.2	26.2	15.0	16.7	17.9	21.9	10.1	4.5	7.0	9.4	16.7
TFRank-4B	SFT	Pointwise	31.4	40.9	19.4	26.2	18.8	19.1	20.3	23.4	13.0	7.7	10.1	9.1	20.0
TFRank-8B	SFT	Pointwise	29.8	**42.3**	**21.5**	25.9	19.7	21.3	22.8	21.6	16.4	6.8	**10.4**	9.0	20.6

Table 1: Main evaluation results (NDCG@10) on the BRIGHT benchmark using BM25 as the retriever. TFRank models are trained on their respective backbones. For each column, the highest value is **bolded** and the second highest is underlined.

rections. The first is to directly exploit the zero-shot reasoning ability of large LLMs (e.g., Llama-3.1-70B (Dubey et al. 2024), GPT-4o), as in JudgeRank (Niu et al. 2024) and InsertRank (Seetharaman, Dhole, and Bansal 2025), which have also motivated some more efficient and effective collaborative multi-agent ranking systems (Fan et al. 2025; Liu et al. 2025b). The second direction equips rankers with reasoning through training. This includes CoT distillation methods such as ReasoningRank (Ji et al. 2024), Rank1 (Weller et al. 2025b), and Rank-K (Yang et al. 2025b), as well as GRPO-based approaches like ReasonRank (Liu et al. 2025a), REARANK (Zhang et al. 2025a), and Rank-R1 (Zhuang et al. 2025), which optimize reasoning-based ranking on lists or sets (Shao et al. 2024). However, most of these methods rely on large-scale models and online CoT generation, leading to high inference costs that limit scalability. In contrast, TFRank employs small-scale LLMs and uses CoT only as a training supervision signal. During inference, it bypasses CoT entirely via a "Think-Free" mechanism, achieving high efficiency while maintaining performance.

Methods

Overall Architecture

TFRank employs a two-stage architecture. In the *training stage*, we instill reasoning abilities into a small-scale LLM (0.5∼8B) through multi-task supervised fine-tuning (SFT). This process leverages a diverse dataset labeled with binary relevance, fine-grained scores, and explicit CoT reasoning. An optional GRPO-based optimization step can also be applied to further enhance the model's performance. During the *inference stage*, inspired by the Qwen3 series (Yang et al. 2025a), we employ a special prompt token as the "think-mode switch" to force the model to bypass explicit CoT generation and emit only a structured pointwise relevance score, thus minimizing latency and output length, making it suitable for real-time ranking scenarios.

Training Data Construction

High-quality training data is the foundation of TFRank's performance. Our data construction involves two steps:

Construction of Foundational Relevance Datasets To equip the model with fundamental ranking abilities, we first construct two datasets containing binary and fine-grained relevance labels, respectively. For **binary relevance data**, following the data processing methodology of Rank1, we acquire approximately 386k positive or negative query-document pairs from the MS MARCO dataset (Nguyen et al. 2016), using only their binary labels; this dataset is denoted as `ms_sub_binary`.

For **fine-grained score data**, we follow the methodology of BGE-M3 (Chen et al. 2024a) to provide the model with

[1]https://github.com/castorini/rank_llm/tree/main/src/rank_llm

Model		Approach	Avg.
BM25	-	Pointwise	36.9
Rank1-7B	SFT	Pointwise	39.2
Rank1-14B	SFT	Pointwise	38.7
REARANK-7B	GRPO	Listwise	42.8
Rank-R1-3B	GRPO	Setwise	37.8
Rank-R1-7B	GRPO	Setwise	<u>43.5</u>
Rank-R1-14B	GRPO	Setwise	**43.8**
Llama3.2 Series			
TFRank-1B	SFT	Pointwise	29.9
TFRank-3B	SFT	Pointwise	40.5
TFRank-8B	SFT	Pointwise	43.2
Qwen2.5 Series			
TFRank-0.5B	SFT	Pointwise	36.6
TFRank-1.5B	SFT	Pointwise	39.5
TFRank-3B	SFT	Pointwise	39.7
TFRank-7B	SFT	Pointwise	41.5
Qwen3 Series			
TFRank-0.6B	SFT	Pointwise	39.0
TFRank-1.7B	SFT	Pointwise	39.4
TFRank-4B	SFT	Pointwise	42.5
TFRank-8B	SFT	Pointwise	42.1

Table 2: Short evaluation results (average NDCG@10) on the BEIR benchmark, using BM25 as the retriever. TFRank models are trained on each corresponding backbone. The best average value for each block is **bolded**, and the second best is <u>underlined</u>. Detailed results for all BEIR datasets are provided in the supplementary material.

more nuanced judgment capabilities. We randomly sample approximately 10% of queries from each MS MARCO length group, along with their top-1 positive and negative documents. For these pairs, we use DeepSeek-R1 to annotate and generate additional documents with fine-grained relevance on a five-point scale. To ensure annotation quality, we apply strict filtering: only positive samples with final scores in $[2, 3, 4]$ and negative samples with scores in $[0, 1]$ are retained. This process yields a fine-grained dataset, denoted as `ms_sub_finegrained`, comprising approximately 7k queries and 44k query-document pairs.

Multi-task Data and CoT Distillation After constructing the `ms_sub_finegrained` dataset, we expand it into three task paradigms: **pointwise**, **pairwise**, and **listwise**. For pointwise data, we prompt DeepSeek-R1 to generate a CoT explanation for each `[query, doc]` pair to justify its known fine-grained score, with the final label appended to the end. For the pairwise setting, we sample `[query,` $\texttt{doc}_1$`,` $\texttt{doc}_2$`]` triplets sharing the same query, and use DeepSeek-R1 to generate a CoT explaining their relative relevance and to produce a preference label. For listwise tasks, we follow the REARANK (Zhang et al. 2025a) format: for each query, we construct a set of documents $D = \{d_1, d_2, ..., d_n\}$, ensuring diversity by sampling no more than two documents per relevance level, and instruct DeepSeek-R1 to generate a CoT that explains the overall ranking logic for the predefined order. This procedure results in a multi-task dataset (`ms_sub_finegrained_cot`) enriched with CoT reasoning signals across all paradigms.

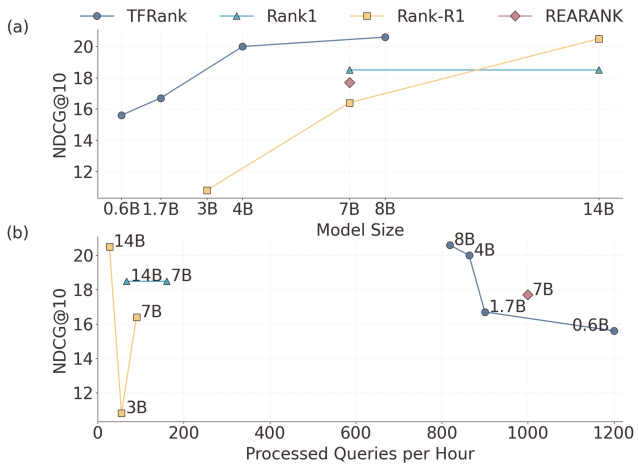

Figure 2: Size and efficiency trade-offs for ranking performance on the BRIGHT benchmark. (a) NDCG@10 versus model size for different ranker families; (b) NDCG@10 versus processed queries per hour (efficiency). All TFRank models are trained on the Qwen3 series.

SFT Training: Internalizing Reasoning

During the SFT stage, our primary goal is to internalize CoT reasoning capabilities into a small-scale model and teach it to switch between `/think` and `/no think` modes based on instructions.

Think-Mode Switch We control the model's generation mode by appending a special token after the input content:

- `/think`: Instructs the model to enter `/think` mode, requiring it to first generate a reasoning process in the format `<think>\n{reasoning text}\n</think>` before providing the final answer.
- `/no think`: Instructs the model to enter `/no think` mode, bypassing the reasoning process entirely and directly outputting the final answer after an empty `<think>\n\n</think>` tag.

Multi-task Supervised Fine-Tuning Based on different task paradigms (listwise, pairwise, and pointwise), constructed training datasets, and the think-mode switch, we construct a unified training data format for SFT:

- `<Instruction><Query><Doc><think mode>`

For clarity, we present below an example of a fine-grained `pointwise` sample in the `/no think` mode:

> **Prompt:** `<Instruct>`: Please judge the relevance strength between the query and the document, and directly output the relevance judgment (yes or no), followed by the relevance score in parentheses, e.g., yes(score) or no(score).
> `<Query>`: what is a stereo preamplifier?

Model	SFT Setting	Inference Mode	StackExchange							Coding		Theorem-based			Avg.
			Bio.	Earth.	Econ.	Psy.	Rob.	Stack.	Sus.	Leet.	Pony	AoPS	TheoQ.	TheoT.	
Qwen3-0.6B	Zero-shot	w/o think	16.6	15.8	5.6	6.4	3.8	4.5	5.7	9.2	2.9	1.7	2.4	2.5	6.4
TFRank-0.6B	full	w/o think	24.8	30.0	12.0	17.5	12.9	12.1	12.7	24.4	13.1	7.1	10.3	9.8	15.6
	full	w/ think	13.9	23.2	9.9	15.1	9.5	8.8	11.7	16.4	6.8	2.6	4.8	3.9	10.5
	w/o RR	w/o think	19.1	25.3	13.8	20.1	11.5	14.7	14.8	23.4	10.8	3.5	8.6	8.7	14.5
	w/o FG	w/o think	25.0	26.3	12.1	20.9	13.2	10.5	15.0	25.7	5.0	6.1	9.5	11.0	15.0
	w/o MT	w/o think	20.5	26.6	10.2	15.8	10.4	9.3	11.0	18.5	8.5	5.6	9.3	10.3	13.0

Table 3: Ablation results (NDCG@10) of TFRank-0.6B (Qwen3) on BRIGHT using BM25 as the retriever. "full" is the complete SFT pipeline; "w/ think" and "w/o think" denote inference with/without explicit reasoning. "w/o RR" is without Relevance Reasoning; "w/o FG" is without Fine-grained Label; "w/o MT" is without Multi-task.

`<Doc>`: Amplifiers are essential components in any sound system, boosting the audio signal to drive loudspeakers and produce audible sound.
`/no think`
Response: `<think>\n\n</think>no(1)`

Through this hybrid training strategy, the model not only learns how to perform explicit reasoning but also learns to decouple its reasoning ability from the final judgment and switch its behavior based on instructions. The training loss for the SFT stage is the standard auto-regressive language model cross-entropy loss:

$$\mathcal{L}_{\text{SFT}} = -\sum_{i=1}^{|T|} \log P(t_i | t_{<i}, C) \tag{1}$$

where C is the input prompt and $T = \{t_1, ..., t_{|T|}\}$ is the target sequence.

Inference: "Think-Free" Pointwise Ranking

During the inference stage, TFRank's design is entirely guided by efficiency and practicality.

Activating `/no think` Mode For all inference requests, we uniformly append the `/no think` command to the prompt. This forces the model to suppress explicit CoT generation, causing it to output only a short answer containing the relevance judgment and score, such as `yes(4)` or `no(1)`. This approach reduces the generation length from hundreds of tokens to just a few, achieving an order-of-magnitude reduction in latency.

Pointwise Score Extraction and Fusion The model's raw output contains two dimensions of signals: a binary judgment (yes/no) and a five-class fine-grained score (0-4). To obtain an expressive final ranking score, we designed a fusion formula. First, we calculate the probability of the binary judgment being "yes", P_{bi}, from the model's output logits:

$$P_{bi} = \text{softmax}([\text{logits}_{\text{yes}}, \text{logits}_{\text{no}}])[0] \tag{2}$$

where $\text{logits}_{\text{yes}}$ and $\text{logits}_{\text{no}}$ are the model's predicted logits for the "yes" and "no" tokens, respectively.

Next, we compute the expected value of the fine-grained score S_{fg}:

$$S_{fg} = \frac{\sum_{i=0}^{4} i \cdot \text{softmax}(\text{logits}_{0...4})}{\max(i) - \min(i)} \tag{3}$$

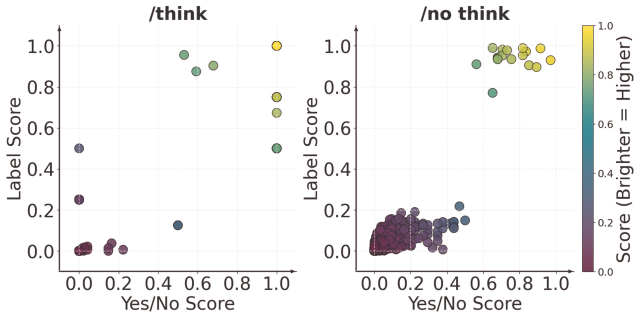

Figure 3: Score distributions for a random 1% sample of BRIGHT, evaluated by TFRank-0.6B (Qwen3) under `/think` and `/no think` inference modes.

where $\text{logits}_{0...4}$ are the model's predicted logits for the five score tokens "0" through "4".

Finally, we average these two scores to obtain the final TFRank sorting score:

$$\text{Score}_{\text{TFRank}} = 0.5 \cdot P_{bi} + 0.5 \cdot S_{fg} \tag{4}$$

Optional: Optimization with GRPO

To further explore the upper bound of relevance modeling in TFRank, we employ GRPO as an advanced optimization strategy. We compare two paradigms: **SFT-then-GRPO**, where the model first undergoes multi-task SFT before further GRPO optimization (labeled as `SFT+GRPO` in Table 4); and **Direct GRPO**, where end-to-end GRPO is directly applied to the instruction-tuned base model.

We design a multi-dimensional reward function R_{total} to optimize both ranking quality and output consistency:

$$R_{\text{total}} = R_{\text{content}} + \lambda R_{\text{format}} \tag{5}$$

where λ is a balancing hyperparameter (default: 0.5).

Content Reward (R_{content}): The content reward is tailored for each task paradigm: for *pointwise* tasks, it averages binary classification accuracy and the mean squared error (MSE) of relevance scores; for *pairwise*, it averages preference accuracy and the MSE scores; For *listwise*, it is the mean of ranking relation accuracy, MSE scores, and relative nDCG improvement (as defined in REARANK).

Model		StackExchange							Coding		Theorem-based			Avg.
		Bio.	Earth.	Econ.	Psy.	Rob.	Stack.	Sus.	Leet.	Pony	AoPS	TheoQ.	TheoT.	
TFRank-0.6B	SFT	24.8	30.0	12.0	17.5	12.9	12.1	12.7	24.4	13.1	7.1	10.3	9.8	15.6
	SFT+GRPO	23.1	28.0	15.1	18.2	11.9	13.8	13.9	23.8	10.6	6.8	8.6	9.9	15.3
	GRPO‡	32.1	43.8	16.4	22.7	23.2	16.9	19.1	25.0	12.9	5.1	8.2	5.8	19.3
TFRank-1.7B	SFT	25.2	29.7	17.2	26.2	15.0	16.7	17.9	21.9	10.1	4.5	7.0	9.4	16.7
	SFT+GRPO	24.9	29.5	16.4	26.2	14.6	15.6	17.5	27.5	9.0	6.5	10.1	9.6	17.3
	GRPO‡	30.9	42.1	19.1	25.4	14.8	21.9	18.8	19.5	15.3	5.3	8.5	10.3	19.3
TFRank-4B	SFT	31.4	40.9	19.4	26.2	18.8	19.1	20.3	23.4	13.0	7.7	10.1	9.1	20.0
	SFT+GRPO	24.8	34.9	16.2	21.2	19.0	19.4	15.7	22.0	11.2	5.1	10.1	9.1	17.4
	GRPO‡	35.3	45.8	21.4	31.1	19.8	24.3	24.9	26.2	16.7	6.8	9.3	11.6	22.8
TFRank-8B	SFT	29.8	42.3	21.5	25.9	19.7	21.3	22.8	21.6	16.4	6.8	10.4	9.0	20.6
	SFT+GRPO	30.3	40.6	21.4	26.1	19.1	20.2	22.5	29.1	14.9	9.8	10.9	10.1	21.3
	GRPO‡	34.5	48.6	23.7	28.3	21.9	22.3	26.8	28.1	21.2	7.2	9.5	11.0	23.6

Table 4: Results (NDCG@10) of TFRank (Qwen3) on BRIGHT under different training paradigms, using BM25 as the retriever. Models marked with ‡ apply GRPO using all training samples, while unmarked models use GRPO on a randomly sampled approximately 20% subset of queries for efficiency.

Format Reward (R_{format}): The format reward enforces strict output conventions, including proper use of the `<think>` tag, minimum reasoning length in `/think` mode, and adherence to the answer format required by each paradigm (e.g., `yes/no(score)` for pointwise).

Experiments

Datasets and Metrics We evaluate TFRank on two representative benchmarks: BRIGHT (Su et al. 2025) and BEIR (Thakur et al. 2021). BRIGHT contains approximately 1,384 real-world queries from diverse domains such as coding, mathematics, and economics, specifically designed to assess multi-step reasoning capabilities beyond keyword matching. BEIR comprises 18 public datasets covering a wide variety of retrieval tasks, including question answering, argument retrieval, fact-checking, biomedical search, and duplicate question detection, emphasizing zero-shot generalization across unseen domains. Following prior work, we primarily report NDCG@10, which effectively measures ranking quality at top positions.

Baselines We compare TFRank with several representative ranking methods. BM25 is included as a classical lexical baseline. We consider zero-shot LLM-based listwise ranking, such as RankGPT (Sun et al. 2023), and recent state-of-the-art methods including Rank1 (Weller et al. 2025b), Rank-R1 (Zhuang et al. 2025), and REARANK (Zhang et al. 2025a). All baselines are evaluated using official implementations or released checkpoints.

Implementation Details TFRank models are trained on three LLM families: Qwen2.5 (0.5B∼7B) (Yang et al. 2024), Qwen3 (0.6B∼8B) (Yang et al. 2025a), and Llama 3.2 (1B∼8B) (Dubey et al. 2024). Llama-3.2-8B is sourced from the open-source community[2] , and the rest from official repositories. Our training setup adopts a learning rate of 1×10^{-5} with random seed 42, and uses at most 5 SFT epochs. In the GRPO training stage, each instance produces eight completions. For checkpoint selection, 1% of the training data is reserved as a validation set. All experiments are performed on one 8-GPU H20 server. More detailed prompts, training data statistics, and full reward definitions are provided in the Appendix and the released code.

[2]https://modelscope.cn/models/voidful/Llama-3.2-8B-Instruct

Results and Analysis

Main Results

Overall Performance As illustrated in Table 1, on the reasoning-intensive BRIGHT benchmark, TFRank **consistently and significantly outperforms** all existing baselines across multiple model families. Notably, TFRank unleashes the potential of small-scale LLMs. For example, Qwen3-based SFT TFRank-1.7B achieves an average NDCG@10 of 16.7, competing with the **4×**-larger Rank1-7B (18.5), REARANK-7B (17.7), and Rank-R1-7B (16.4). Similarly, TFRank-3B, based on Llama-3.2 and Qwen-2.5, even outperforms some aforementioned 7B baselines. On the general-purpose BEIR benchmark, in Table 2, TFRank delivers performance on par with the strongest baselines. For instance, TFRank-8B, based on Llama-3.2, achieves an average NDCG@10 of 43.2, outperforming Rank1-14B and REARANK-7B, while remaining comparable to the 43.5 and 43.8 achieved by Rank-R1-7B and Rank-R1-14B.

Model Efficiency While achieving high performance, TFRank demonstrates **significant efficiency advantages**. As shown in Figure 2(a), on the BRIGHT benchmark, the TFRank performance curve is consistently above other models, indicating its ability to achieve superior performance at equivalent or even smaller model scales, which is crucial for resource-constrained deployment scenarios. Figure 2(b) further reveals its inference efficiency. By adopting a pointwise scoring and "Think-Free" reasoning mode, TFRank processes far more queries per hour than methods that re-

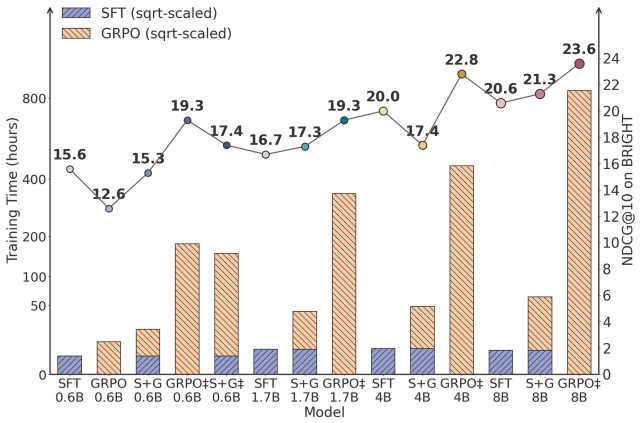

Figure 4: Training time (hours) versus ranking performance (NDCG@10 on BRIGHT) for different training strategies. All TFRank models use the Qwen3 series backbone. The meaning of the ‡ symbol follows that in Table 4.

quire explicit CoT generation, such as Rank1 and Rank-R1. Compared to the listwise model REARANK, TFRank's pointwise scoring enables intra-query parallelism, allowing single-query throughput to scale effectively with hardware resources. This is particularly advantageous for latency-sensitive applications. Remarkably, the 0.6B TFRank surpasses the 7B REARANK in performance on BRIGHT, while offering a significant throughput advantage.

Ablation Study

We conducted a series of ablation studies on the TFRank-0.6B (Qwen3) model, with the results presented in Table 3.

Effectiveness of Components Our full TFRank model achieves a performance of 15.6, more than doubling the score attained by its base model and demonstrating the overall effectiveness of our training method. Ablation experiments confirm that each of our proposed components is indispensable: removing either Relevance Reasoning Supervision, Fine-grained Relevance Labels, or Multi-task training leads to a clear performance decline.

"Think-Free" Reasoning A striking finding is the effectiveness of "Think-Free" reasoning. Forcing explicit CoT at inference ("w/ think") severely damages performance, reducing NDCG@10 to 10.5. As illustrated in Figure 3, we observe that the score distribution in the `/think` mode is more concentrated, indicating a weaker capacity to discern documents with partial relevance. This aligns with observations by Jedidi et al. (2025): explicit reasoning can induce overconfidence and impair the model's ability to model partial relevance finely. By internalizing the reasoning process during training, TFRank can output more precise and discriminative scores at inference time without explicit "thinking", thereby achieving a substantial increase in efficiency.

Further Optimization with GRPO

To explore the performance ceiling of TFRank, we introduced the GRPO training strategy. As shown in Table 4, 1.7B and 8B models show small but consistent gains when GRPO is applied after SFT. Interestingly, across the full Qwen3 series, directly applying full-data GRPO yields significant improvements over SFT or SFT+GRPO. This may be because Qwen3 models, as reasoning-oriented LLMs, can better exploit GRPO's broader policy exploration to discover more optimal reasoning and ranking strategies than standard SFT. However, this performance improvement comes at a significant training cost. As depicted in Figure 4, the training duration for GRPO far exceeds that of SFT. For example, the cost of full GRPO training on the 0.6B model (approx. 200 hours) is much higher than that of training a superior-performing 8B SFT model (approx. 8 hours). Therefore, while increasing model scale via SFT or increasing the amount of GRPO data can both effectively boost performance, they represent a trade-off between computational resources and time.

BRIGHT Leaderboard and Domain Generalization

To validate TFRank's competitiveness within state-of-the-art retrieval systems, we employ ReasonIR-8B (Shao et al. 2025) as the retriever and compare both our models and baselines against top BRIGHT leaderboard entries. To further examine TFRank's generalization ability on reasoning-intensive ranking, we evaluate it on a domain-specific dataset, R2MED (Li, Zhou, and Liu 2025), which focuses on complex medical retrieval, using E5-mistral-7b-instruct (Wang et al. 2023) as the retriever. Detailed results are provided in the Appendix. Experimental results show that GRPO TFRank-8B achieves an NDCG@10 of 32.0 on BRIGHT, substantially outperforming REARANK-7B (24.2) and Rank-R1-7B (23.2), while remaining slightly behind much larger models such as Rank-R1-32B. On R2Med, TFRank reaches an NDCG@10 of 34.7, surpassing strong baselines such as Rank-R1-7B (32.87) and Rank1-7B (32.30). These results demonstrate that TFRank is a competitive and generalizable alternative when deploying ultra-large models is not feasible.

Conclusion

We introduce TFRank, an innovative framework that addresses the efficiency bottlenecks of reasoning-intensive rankers in real-world deployments. Through a "Think-Free" reasoning mechanism, TFRank uses multi-task learning and CoT supervision during training to internalize complex reasoning capabilities within small-scale LLMs. During inference, it completely bypasses the generation of explicit reasoning chains, directly and efficiently producing pointwise relevance scores. Experiments demonstrate that this approach enables small-scale models (e.g., 1.7B) to compete with baselines with over four times the parameters, while substantially reducing inference latency. We further observe that once reasoning is internalized, suppressing CoT generation during inference yields more accurate ranking results. In summary, TFRank provides a practical path for building and deploying advanced reasoning-based rankers, offering a feasible solution to popularize complex AI capabilities in resource-constrained applications.

Acknowledgments

We would like to express our sincere gratitude to the anonymous reviewers for their valuable feedback. We also thank the Chairs and the organizing staff for their dedicated efforts in facilitating this work. This work is supported by the National Natural Science Foundation of China (62272439/62572456) and the Fundamental Research Funds for the Central Universities.

References

Chen, J.; Xiao, S.; Zhang, P.; Luo, K.; Lian, D.; and Liu, Z. 2024a. BGE M3-Embedding: Multi-Lingual, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation. *CoRR*, abs/2402.03216.

Chen, X.; He, B.; Lin, H.; Han, X.; Wang, T.; Cao, B.; Sun, L.; and Sun, Y. 2024b. Spiral of Silence: How is Large Language Model Killing Information Retrieval? - A Case Study on Open Domain Question Answering. In Ku, L.; Martins, A.; and Srikumar, V., eds., *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, 14930–14951. Association for Computational Linguistics.

DeepSeek-AI; Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; Bi, X.; Zhang, X.; Yu, X.; Wu, Y.; Wu, Z. F.; Gou, Z.; Shao, Z.; Li, Z.; Gao, Z.; Liu, A.; Xue, B.; Wang, B.; Wu, B.; Feng, B.; Lu, C.; Zhao, C.; Deng, C.; Zhang, C.; Ruan, C.; Dai, D.; Chen, D.; Ji, D.; Li, E.; Lin, F.; Dai, F.; Luo, F.; Hao, G.; Chen, G.; Li, G.; Zhang, H.; Bao, H.; Xu, H.; Wang, H.; Ding, H.; Xin, H.; Gao, H.; Qu, H.; Li, H.; Guo, J.; Li, J.; Wang, J.; Chen, J.; Yuan, J.; Qiu, J.; Li, J.; Cai, J. L.; Ni, J.; Liang, J.; Chen, J.; Dong, K.; Hu, K.; Gao, K.; Guan, K.; Huang, K.; Yu, K.; Wang, L.; Zhang, L.; Zhao, L.; Wang, L.; Zhang, L.; Xu, L.; Xia, L.; Zhang, M.; Zhang, M.; Tang, M.; Li, M.; Wang, M.; Li, M.; Tian, N.; Huang, P.; Zhang, P.; Wang, Q.; Chen, Q.; Du, Q.; Ge, R.; Zhang, R.; Pan, R.; Wang, R.; Chen, R. J.; Jin, R. L.; Chen, R.; Lu, S.; Zhou, S.; Chen, S.; Ye, S.; Wang, S.; Yu, S.; Zhou, S.; Pan, S.; and Li, S. S. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *CoRR*, abs/2501.12948.

Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Yang, A.; Fan, A.; et al. 2024. The llama 3 herd of models. *arXiv e-prints*, arXiv–2407.

Fan, Y.; Xue, K.; Li, Z.; Zhang, X.; and Ruan, T. 2025. An LLM-based Framework for Biomedical Terminology Normalization in Social Media via Multi-Agent Collaboration. In *Proceedings of the 31st International Conference on Computational Linguistics*, 10712–10726.

Guo, T.; Chen, X.; Wang, Y.; Chang, R.; Pei, S.; Chawla, N. V.; Wiest, O.; and Zhang, X. 2024. Large Language Model Based Multi-agents: A Survey of Progress and Challenges. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024*, 8048–8057. ijcai.org.

Jedidi, N.; Chuang, Y.; Glass, J. R.; and Lin, J. 2025. Don't "Overthink" Passage Reranking: Is Reasoning Truly Necessary? *CoRR*, abs/2505.16886.

Ji, Y.; Li, Z.; Meng, R.; and He, D. 2024. ReasoningRank: Teaching Student Models to Rank through Reasoning-Based Knowledge Distillation. *CoRR*, abs/2410.05168.

Li, L.; Zhou, X.; and Liu, Z. 2025. R2MED: A Benchmark for Reasoning-Driven Medical Retrieval. *arXiv preprint arXiv:2505.14558*.

Li, X.; Shakir, A.; Huang, R.; Lipp, J.; and Li, J. 2025. ProRank: Prompt Warmup via Reinforcement Learning for Small Language Models Reranking. *CoRR*, abs/2506.03487.

Liu, W.; Ma, X.; Sun, W.; Zhu, Y.; Li, Y.; Yin, D.; and Dou, Z. 2025a. ReasonRank: Empowering Passage Ranking with Strong Reasoning Ability. *arXiv preprint arXiv:2508.07050*.

Liu, W.; Ma, X.; Zhu, Y.; Su, L.; Wang, S.; Yin, D.; and Dou, Z. 2025b. CoRanking: Collaborative Ranking with Small and Large Ranking Agents. *arXiv preprint arXiv:2503.23427*.

Liu, W.; Ma, X.; Zhu, Y.; Zhao, Z.; Wang, S.; Yin, D.; and Dou, Z. 2025c. Sliding Windows Are Not the End: Exploring Full Ranking with Long-Context Large Language Models. In Che, W.; Nabende, J.; Shutova, E.; and Pilehvar, M. T., eds., *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 162–176. Vienna, Austria: Association for Computational Linguistics. ISBN 979-8-89176-251-0.

Liu, W.; Zhu, Y.; and Dou, Z. 2024. Demorank: Selecting effective demonstrations for large language models in ranking task. *arXiv preprint arXiv:2406.16332*.

Luo, J.; Chen, X.; He, B.; and Sun, L. 2024. PRP-Graph: Pairwise Ranking Prompting to LLMs with Graph Aggregation for Effective Text Re-ranking. In Ku, L.; Martins, A.; and Srikumar, V., eds., *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, 5766–5776. Association for Computational Linguistics.

Ma, X.; Wang, L.; Yang, N.; Wei, F.; and Lin, J. 2024. Fine-Tuning LLaMA for Multi-Stage Text Retrieval. In Yang, G. H.; Wang, H.; Han, S.; Hauff, C.; Zuccon, G.; and Zhang, Y., eds., *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024*, 2421–2425. ACM.

Ma, X.; Zhang, X.; Pradeep, R.; and Lin, J. 2023. Zero-Shot Listwise Document Reranking with a Large Language Model. *CoRR*, abs/2305.02156.

Nguyen, T.; Rosenberg, M.; Song, X.; Gao, J.; Tiwary, S.; Majumder, R.; and Deng, L. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. In Besold, T. R.; Bordes, A.; d'Avila Garcez, A. S.; and Wayne, G., eds., *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona,*

Spain, December 9, 2016, volume 1773 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Niu, T.; Joty, S.; Liu, Y.; Xiong, C.; Zhou, Y.; and Yavuz, S. 2024. JudgeRank: Leveraging Large Language Models for Reasoning-Intensive Reranking. *CoRR*, abs/2411.00142.

Qin, Z.; Jagerman, R.; Hui, K.; Zhuang, H.; Wu, J.; Yan, L.; Shen, J.; Liu, T.; Liu, J.; Metzler, D.; Wang, X.; and Bendersky, M. 2024. Large Language Models are Effective Text Rankers with Pairwise Ranking Prompting. In Duh, K.; Gómez-Adorno, H.; and Bethard, S., eds., *Findings of the Association for Computational Linguistics: NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, 1504–1518. Association for Computational Linguistics.

Seetharaman, R.; Dhole, K. D.; and Bansal, A. 2025. InsertRank: LLMs can reason over BM25 scores to Improve Listwise Reranking. *CoRR*, abs/2506.14086.

Shao, R.; Qiao, R.; Kishore, V.; Muennighoff, N.; Lin, X. V.; Rus, D.; Low, B. K. H.; Min, S.; Yih, W.; Koh, P. W.; and Zettlemoyer, L. 2025. ReasonIR: Training Retrievers for Reasoning Tasks. *CoRR*, abs/2504.20595.

Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y.; et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Su, H.; Yen, H.; Xia, M.; Shi, W.; Muennighoff, N.; Wang, H.; Liu, H.; Shi, Q.; Siegel, Z. S.; Tang, M.; Sun, R.; Yoon, J.; Arik, S. Ö.; Chen, D.; and Yu, T. 2025. BRIGHT: A Realistic and Challenging Benchmark for Reasoning-Intensive Retrieval. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Sun, W.; Yan, L.; Ma, X.; Wang, S.; Ren, P.; Chen, Z.; Yin, D.; and Ren, Z. 2023. Is ChatGPT Good at Search? Investigating Large Language Models as Re-Ranking Agents. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, 14918–14937. Association for Computational Linguistics.

Thakur, N.; Reimers, N.; Rücklé, A.; Srivastava, A.; and Gurevych, I. 2021. BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models. *CoRR*, abs/2104.08663.

Wang, L.; Yang, N.; Huang, X.; Yang, L.; Majumder, R.; and Wei, F. 2023. Improving Text Embeddings with Large Language Models. *arXiv preprint arXiv:2401.00368*.

Weller, O.; Chang, B.; MacAvaney, S.; Lo, K.; Cohan, A.; Durme, B. V.; Lawrie, D. J.; and Soldaini, L. 2025a. FollowIR: Evaluating and Teaching Information Retrieval Models to Follow Instructions. In Chiruzzo, L.; Ritter, A.; and Wang, L., eds., *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2025 - Volume 1: Long Papers, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, 11926–11942. Association for Computational Linguistics.

Weller, O.; Ricci, K.; Yang, E.; Yates, A.; Lawrie, D.; and Van Durme, B. 2025b. Rank1: Test-time compute for reranking in information retrieval. *arXiv preprint arXiv:2502.18418*.

Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Gao, C.; Huang, C.; Lv, C.; et al. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Yang, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Li, C.; Liu, D.; Huang, F.; Wei, H.; Lin, H.; Yang, J.; Tu, J.; Zhang, J.; Yang, J.; Yang, J.; Zhou, J.; Lin, J.; Dang, K.; Lu, K.; Bao, K.; Yang, K.; Yu, L.; Li, M.; Xue, M.; Zhang, P.; Zhu, Q.; Men, R.; Lin, R.; Li, T.; Xia, T.; Ren, X.; Ren, X.; Fan, Y.; Su, Y.; Zhang, Y.; Wan, Y.; Liu, Y.; Cui, Z.; Zhang, Z.; and Qiu, Z. 2024. Qwen2.5 Technical Report. *CoRR*, abs/2412.15115.

Yang, E.; Yates, A.; Ricci, K.; Weller, O.; Chari, V.; Durme, B. V.; and Lawrie, D. J. 2025b. Rank-K: Test-Time Reasoning for Listwise Reranking. *CoRR*, abs/2505.14432.

Yu, Y.; Ping, W.; Liu, Z.; Wang, B.; You, J.; Zhang, C.; Shoeybi, M.; and Catanzaro, B. 2024. RankRAG: Unifying Context Ranking with Retrieval-Augmented Generation in LLMs. In Globersons, A.; Mackey, L.; Belgrave, D.; Fan, A.; Paquet, U.; Tomczak, J. M.; and Zhang, C., eds., *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Zhang, L.; Wang, B.; Qiu, X.; Reddy, S.; and Agrawal, A. 2025a. REARANK: Reasoning Re-ranking Agent via Reinforcement Learning. *arXiv preprint arXiv:2505.20046*.

Zhang, Y.; Li, M.; Long, D.; Zhang, X.; Lin, H.; Yang, B.; Xie, P.; Yang, A.; Liu, D.; Lin, J.; Huang, F.; and Zhou, J. 2025b. Qwen3 Embedding: Advancing Text Embedding and Reranking Through Foundation Models. *CoRR*, abs/2506.05176.

Zhu, Y.; Yuan, H.; Wang, S.; Liu, J.; Liu, W.; Deng, C.; Chen, H.; Liu, Z.; Dou, Z.; and Wen, J.-R. 2023. Large language models for information retrieval: A survey. *arXiv preprint arXiv:2308.07107*.

Zhuang, S.; Ma, X.; Koopman, B.; Lin, J.; and Zuccon, G. 2025. Rank-r1: Enhancing reasoning in llm-based document rerankers via reinforcement learning. *arXiv preprint arXiv:2503.06034*.

Zhuang, S.; Zhuang, H.; Koopman, B.; and Zuccon, G. 2024. A Setwise Approach for Effective and Highly Efficient Zero-shot Ranking with Large Language Models. In Yang, G. H.; Wang, H.; Han, S.; Hauff, C.; Zuccon, G.; and Zhang, Y., eds., *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024*, 38–47. ACM.