ERANK: Fusing Supervised Fine-Tuning and Reinforcement Learning for Effective and Efficient Text Reranking

Yuzheng Cai[1], **Yanzhao Zhang**[2], **Dingkun Long**[2], **Mingxin Li**[2], **Pengjun Xie**[2], **Weiguo Zheng**[1] ✉

[1]School of Data Science, Fudan University
[2]Alibaba Group
yuzhengcai21@m.fudan.edu.cn, zhengweiguo@fudan.edu.cn

Abstract

Text reranking models are a crucial component in modern systems like Retrieval-Augmented Generation, tasked with selecting the most relevant documents prior to generation. However, current Large Language Models (LLMs) powered rerankers often face a fundamental trade-off. On one hand, Supervised Fine-Tuning based pointwise methods that frame relevance as a binary classification task lack the necessary scoring discrimination, particularly for those built on reasoning LLMs. On the other hand, approaches designed for complex reasoning often employ powerful yet inefficient listwise formulations, rendering them impractical for low latency applications. To resolve this dilemma, we introduce ERANK, a highly *Effective* and *Efficient* pointwise reranker built from a reasoning LLM that excels across diverse relevance scenarios. We propose a novel two-stage training pipeline that begins with Supervised Fine-Tuning (SFT). In this stage, we move beyond binary labels and train the model generatively to output fine grained integer scores, which significantly enhances relevance discrimination. The model is then further refined using Reinforcement Learning (RL) with a novel, listwise derived reward. This technique instills global ranking awareness into the efficient pointwise architecture. We evaluate the ERANK reranker on the BRIGHT, FollowIR, TREC DL, and BEIR benchmarks, demonstrating superior effectiveness and robustness compared to existing approaches. On the reasoning-intensive BRIGHT benchmark, our ERANK-4B achieves an nDCG@10 of 38.7, while a larger 32B variant reaches a state of the art nDCG@10 of 40.2.

HuggingFace — https://huggingface.co/collections/Alibaba-NLP/erank

ModelScope — https://modelscope.cn/collections/ERank-488bc43c873e4c

Extended version — https://arxiv.org/abs/2509.00520

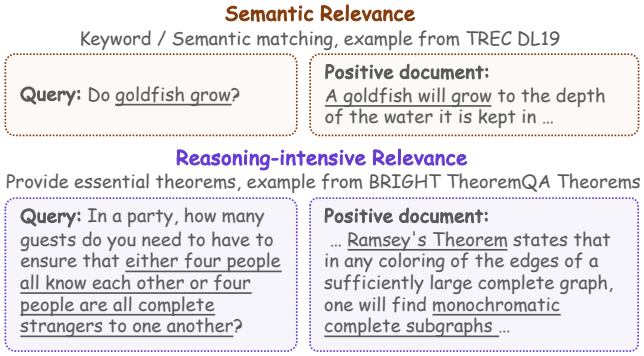

Figure 1: Semantic relevance refers to the traditional understanding based on keyword or semantic matching, while the reasoning-intensive example aims to capture documents that may not directly answer the query but provide essential intermediate information needed for multi-step reasoning.

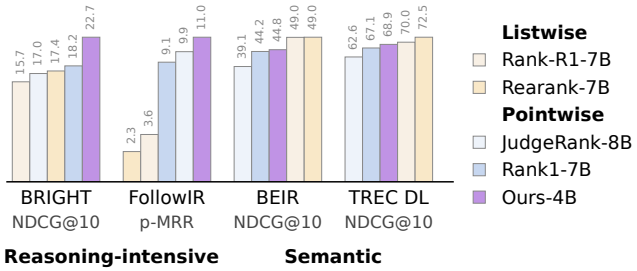

Figure 2: ERANK-4B achieves state-of-the-art performance among pointwise rerankers using candidate documents retrieved by BM25 with original queries. Under retrieval settings in Section 4.2, ERANK-4B and 32B further achieve the nDCG@10 of 38.7 and 40.2 on BRIGHT, respectively.

1 Introduction

Text reranking is a fundamental component of various Natural Language Processing and Information Retrieval applications, utilized extensively in downstream tasks such as open-domain question answering (Lee et al. 2018), web search (Lin, Nogueira, and Yates 2022), and recommendation systems (Chuang et al. 2020; Gao et al. 2025). Large Language Models (LLMs) have significantly reshaped the text reranking landscape. On one hand, studies have sought to leverage the advanced text understanding capabilities of LLMs for reranking, either through zero-shot prompting or Supervised Fine-Tuning (Zhang et al. 2023; Liu et al. 2024). On the other hand, LLMs have introduced new application paradigms like Retrieval-Augmented Generation (Wu et al. 2024; Gupta, Ranjan, and Singh 2024; Wang et al. 2024) and

agentic systems (Huang et al. 2024; Li et al. 2024). These paradigms demand capabilities beyond traditional semantic relevance, requiring models to perform reasoning-intensive retrieval, such as identifying issue-relevant code snippets to resolve a specific programming problem. Recent advancements in test-time compute (OpenAI 2024; DeepSeek AI 2025) have shown promise in such scenarios, with a growing number of text rerankers based on reasoning LLMs (Weller et al. 2025b; Zhuang et al. 2025; Zhang et al. 2025).

Prior approaches have generally treated traditional semantic relevance and reasoning-intensive reranking as distinct challenges, which are illustrated in Figure 1. For semantic tasks, Supervised Fine-Tuning (SFT) is a common strategy (Ma et al. 2024; Sun et al. 2023; Zhang et al. 2024). However, most SFT-based rerankers adopt a pointwise scoring method based on binary classification, where the model predicts labels like "Relevant" or "Not Relevant". We argue this approach is suboptimal as it leads to poor score discrimination, a problem exacerbated in modern reasoning LLMs that generate overconfident predictions after Chain-of-Thought (CoT). For reasoning-intensive tasks, Reinforcement Learning (RL) has shown promise (Zhuang et al. 2025; Zhang et al. 2025). However, these methods often rely on listwise or setwise formulations and ingest multiple candidate documents simultaneously. With sliding windows, they process different batches of documents sequentially, resulting in prohibitive latency and memory footprints that make them impractical for real-world deployment.

This work addresses a central question: can a single, efficient reranker powered by reasoning LLM be trained to excel at both semantic relevance and deep reasoning? We contend that this is achievable by enhancing the pointwise architecture, which scores each document independently. We introduce a novel, two-stage training framework illustrated in Figure 4, which seamlessly integrates Supervised Fine-Tuning (SFT) with Reinforcement Learning (RL) for LLM-based reranker training. The first stage, SFT, trains a base model on a diverse mixture of semantic and reasoning-oriented data. Crucially, we abandon the standard binary classification paradigm and instead train the model using a fine-grained integer scoring scheme, which fully utilizes the generative power of LLMs and significantly improves score discrimination. We also employ a data synthesis strategy to generate high-quality reasoning chains and fine-grained scores to overcome data scarcity. In the second stage, we further refine the SFT-tuned model using RL. To bridge the gap between listwise optimality and pointwise efficiency, we introduce a novel, listwise-derived reward function. This function provides a global ranking signal during training, encouraging the model to learn the relative importance of documents. This allows our pointwise model to benefit from listwise-style optimization while retaining its low latency.

Extensive experiments on semantic (TREC DL (Craswell et al. 2020, 2021), BEIR (Thakur et al. 2021)) and reasoning-intensive benchmarks (BRIGHT (Su et al. 2024), FollowIR (Weller et al. 2025a)) confirm that our framework delivers substantial gains. As shown in Figure 2, our 4B-parameter model outperforms many 7B model size rerankers, and our 32B model sets a new state-of-the-art

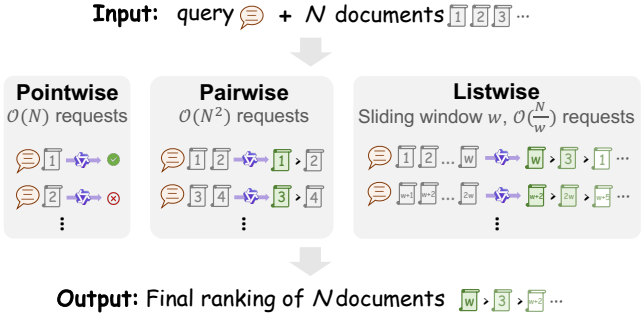

Figure 3: Comparison of different reranking paradigms.

on the BRIGHT benchmark. Latency measurements confirm our models maintain the high efficiency of standard pointwise rerankers, making them both powerful and practical.

Briefly, our main contributions are to:

- Reveal the suboptimality of binary classification for LLM rerankers and propose a generative approach that outputs discrete integer scores to enhance score discrimination.
- Introduce a novel two-stage framework integrating Supervised Fine-Tuning (SFT) and Reinforcement Learning (RL) to build a single, efficient pointwise reranker for both semantic and reasoning-intensive tasks.
- Our model, ERANK, sets a new state-of-the-art on the reasoning-based BRIGHT benchmark while demonstrating exceptional performance on standard semantic tasks.

2 Related Work

LLM for Text Reranking. Large Language Models (LLMs) have significantly advanced text reranking beyond the capabilities of earlier encoder-only models such as BERT (Liu et al. 2024). LLMs are typically applied to this task using either zero-shot prompting (Zhang et al. 2023; Zhuang et al. 2024; Niu et al. 2024) or, more effectively, Supervised Fine-Tuning (Ma et al. 2024; Zhang et al. 2024). As shown in Figure 3, reranking methodologies are broadly categorized into pointwise (Liang et al. 2022), pairwise (Qin et al. 2024), and listwise approaches (Ma et al. 2023; Sun et al. 2023). Listwise methods, which evaluate a list of candidate documents, generally yield the highest ranking quality by directly optimizing the document order (Gao, Dai, and Callan 2021; Zhang et al. 2022; Liu et al. 2025). However, their computational cost scales quadratically with input length, making them impractical for real-world systems that demand low latency. In contrast, pointwise methods score each query-document pair independently. This paradigm enables massive parallelization and efficient inference, establishing it as the preferred choice for large scale deployment. Most fine-tuned pointwise rerankers conventionally treat the task as a binary classification problem. We argue this approach fails to leverage the full generative power of modern LLMs and results in suboptimal performance.

Reinforcement Learning for Reranking. The success of Reinforcement Learning (RL) in enhancing the complex reasoning abilities of LLMs, such as OpenAI-O1 (OpenAI 2024) and DeepSeek-R1 (DeepSeek AI 2025), has inspired

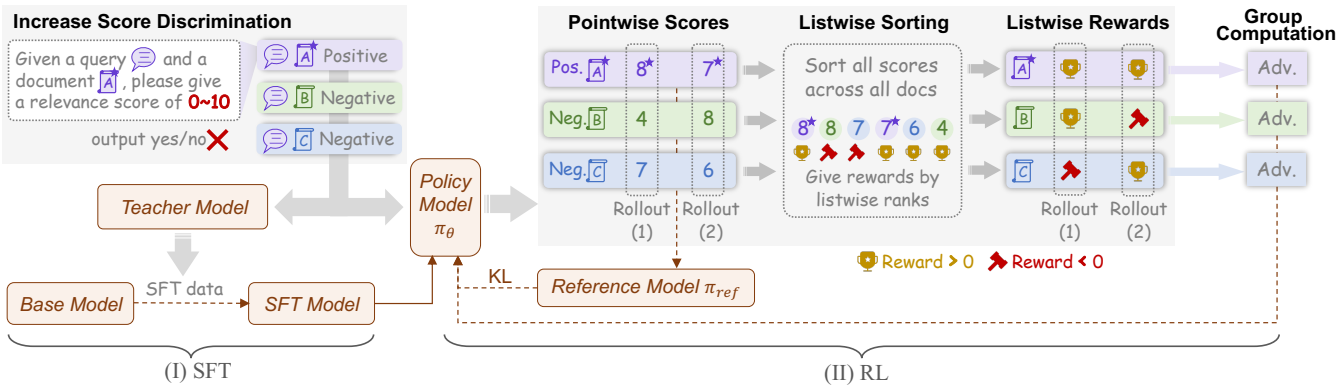

Figure 4: Overview of the two-stage fine-tuning pipeline for the pointwise ERANK reranker. Given a query and $N = 3$ documents A, B and C, where A is the only positively related one, the SFT model is trained to deliver a relevance score ranging from 0 to 10 for each document. During RL training, the model generates $G = 2$ rollouts for each document. These $N \times G = 6$ scores extracted from all rollouts across all documents are then sorted together to compute listwise ranking derived rewards.

its application to reranking. Recent work demonstrates that RL can refine a model's capacity to identify documents that are not merely semantically relevant but also instrumentally useful for resolving a user's query. However, these pioneering RL-based reranking methods predominantly adopt listwise or setwise training frameworks (Zhuang et al. 2025; Zhang et al. 2025). While effective, they inherit the high latency and memory requirements associated with processing multiple batches of documents sequentially, which limits their practical applicability. Our work addresses this critical gap by improving pointwise relevance discrimination with a generative SFT stage using fine-grained scores. Then, we employ RL to optimize our pointwise model with a globally aware listwise reward signal, which achieves the ranking quality of listwise methods while preserving the inference efficiency of a pointwise architecture.

3 Method

Our training methodology unfolds in a two-stage pipeline designed to build a reranker that excels at both semantic relevance and reasoning-intensive relevance. The first stage uses Supervised Fine-Tuning (SFT) to establish a strong foundation, and the second stage employs Reinforcement Learning (RL) with the Group Relative Policy Optimization (GRPO) algorithm (Shao et al. 2024) to refine the reranking ability.

3.1 Task Formulation

We formulate the text reranking task as a generative process. With a specific instruction I that defines the relevance criteria, given a query q and a set of N candidate documents $\{d_1, d_2, \ldots, d_N\}$, our model processes each query-document pair independently. For each pair, it generates a response that includes a Chain-of-Thought (CoT) c_i explaining its reasoning, followed by a relevance score s_i. This is represented by the conditional probability of policy LLM π:

$$\pi(c_i, s_i \mid I, q, d_i), \quad i = 1, 2, \ldots, N.$$

Based on the extracted scores $\{s_1, s_2, \ldots, s_N\}$, the documents are then sorted in descending order to produce the final ranked list. This pointwise formulation ensures low inference latency and provides interpretability through the generated reasoning Chain-of-Thought (CoT).

3.2 Supervised Fine-Tuning with Fine-Grained Scores

To perform Supervised Fine-Tuning (SFT), we construct a dataset $\mathcal{D} = \{(x_k, y_k)\}_{k=1}^{K}$, where each input x_k is a prompt containing the relevance instruction I, a query q, and a document d. The target output y_k is a combination of reasoning chain c and relevance score s, as formatted in Figure 5.

Generative Fine-Grained Scoring. A central limitation of prior pointwise rerankers is their reliance on a binary classification objective, where the model is trained to predict labels of *"yes"* and *"no"* to represent relevance. To leverage the generative nature of LLMs, some recent methods compute a normalized relevance score by extracting token probabilities for *"yes"* and *"no"*, which can be defined as:

$$\frac{\Pr(\text{token}=yes)}{\Pr(\text{token}=yes) + \Pr(\text{token}=no)}.$$

Our experiments reveal that such a strategy leads to poor score discrimination. This issue is particularly pronounced in reasoning LLMs, where the tendency inherent in Chain-of-Thought (CoT) reasoning leads to overconfident predictions. As illustrated in Figure 6 and detailed in Appendix B, a comparison between a non-reasoning model (Qwen3-32B (Qwen Team 2025a)) and a reasoning-enhanced model (QwQ-32B (Qwen Team 2025b)) shows that the latter produces normalized scores heavily concentrated near 0 or 1. A significantly higher proportion of scores from the reasoning model falls within the extreme intervals of $[0, 0.00001]$ and $[0.99999, 1]$. This concentration severely diminishes the model's ability to distinguish between varying degrees of relevance, which is essential for effective reranking.

To overcome this limitation, we reframe reranking as a generative task with a fine-grained scoring system. Instead of predicting binary labels, we train the model to generate

Given a query and a document, please give a relevance score of 0 to 10. The goal or relevance definition is: {instruction}

Here is the query:
{query}

Here is the document:
{document}

After thinking, directly choose a relevance score from [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10].
- 0 represents completely not related.
- 10 means perfectly related.

Desired output format:
<think>put your thinking here</think><answer>Only allows an integer here</answer>

Your output:

Figure 5: Prompt for scoring with integers from 0 to 10.

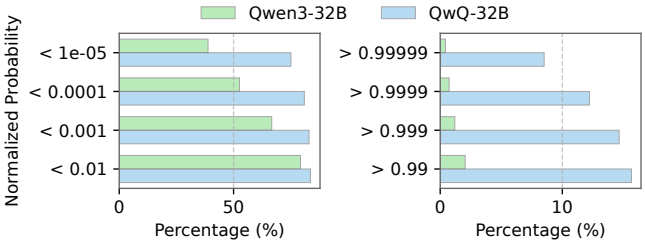

Figure 6: Distributions of normalized probability with non-reasoning and reasoning LLMs on BRIGHT benchmark.

Scoring	BRIGHT	TREC DL	BEIR (5 subsets)
yes / no	20.8	60.9	31.1
$\{0, 1, 2, 3\}$	22.7	64.1	36.5
$\{0, 1, \cdots, 10\}$	23.2	66.1	37.1

Table 1: Average nDCG@10 of QwQ-32B reasoning LLM when varying scoring discrimination on three benchmarks.

an integer score from 0 to 10 that reflects the degree of relevance with prompt in Figure 5. Then, the final ranking score is computed as $s_i \times \text{Pr}(\text{token}{=}s_i)$. This method fully utilizes the autoregressive capabilities of the LLM, creating a more expressive and discriminative scoring space critical for distinguishing between documents of varying quality. Table 1 shows that it consistently improves nDCG@10 across multiple benchmarks under experimental settings in Appendix B.

Data Synthesis for SFT. To train our model for this fine-grained scoring task, we synthesize a high-quality dataset that covers both semantic matching and complex reasoning scenarios. We employ a powerful open-source model, QwQ-32B (Qwen Team 2025b), as a teacher to generate reasoning chains and integer scores. To ensure the quality and reliability of the synthetic labels, our data construction process emphasizes two key aspects.

(1) Query-Document Diversity: We source query-document pairs from a diverse mix of datasets, including MS MARCO (Bajaj et al. 2016) for semantic relevance, and ReasonIR (Shao et al. 2025) and Promptriever (Weller et al. 2024) for complex reasoning tasks. For semantic relevance, we randomly select $5,000$ queries from MS MARCO. For reasoning-intensive tasks, we sample $10,000$ queries from the hard query (HQ) set of ReasonIR and $5,000$ queries from the Promptriever training set. Both of these sources contain complex queries that require deep reasoning. For each query, we enrich the initial candidate pool, which includes the annotated positive documents and synthetic negative documents, by retrieving the top $1,000$ documents from the corpus using the ReasonIR-8B retriever. We then sample documents from different ranking ranges to create a balanced set of negatives: the top 10 documents serve as hard negatives, positions 11–100 as medium negatives, and positions 101–$1,000$ as easy negatives. Further details are provided in Appendix C. Each query is ultimately associated with exactly 20 documents for a consistent input structure.

(2) High-Quality and Stable Reasoning Trajectory Generation: We use QwQ-32B as the teacher to generate a reasoning chain c and a corresponding score s for each query-document pair. To improve the reliability of these generated labels, we perform multiple independent generations for each pair and compute the average score, which serves as a consensus score. We then select the single generation whose score is closest to this consensus. Experiments on a random sample of 512 queries show that it improves the average nDCG@10 from 63.5% with a single generation to 65.6% with 3-sample consensus and further to 67.4% with 10-sample consensus. To balance performance and costs, we use 3 generations per instance in our final data synthesis process. Finally, we filter out any instances where the generated output exceeds $2,048$ tokens. This procedure results in our final Supervised Fine-Tuning (SFT) dataset, $\mathcal{D}$.

Such a high-quality dataset $\mathcal{D}$ enables the model to learn nuanced relevance assessment. The model is then trained on this dataset using a standard language modeling objective:

$$\mathcal{L}_{SFT}(\theta) = -\sum_{(x,y)\in\mathcal{D}} \log P(y \mid x; \theta).$$

3.3 Reinforcement Learning with GRPO

While Supervised Fine-Tuning (SFT) provides a strong reranking model, we employ Reinforcement Learning (RL) to further refine its ability to discern subtle ranking differences and optimize for list level metrics. To achieve this, we adopt the Group Relative Policy Optimization (GRPO) algorithm (Shao et al. 2024), inspired by prior work demonstrating that RL on small, high quality datasets can yield significant performance gains (DeepSeek AI 2025). We initialize both the GRPO policy π_θ and the reference model π_{ref} with the SFT-tuned model to ensure training stability and preserve its well generalized capabilities.

The training process begins by sampling a group of G output trajectories $\{y_1, y_2, \ldots, y_G\}$ for each input prompt with the old policy π_{old}. The policy π_θ is then updated by optimizing the GRPO objective. This objective is built around a

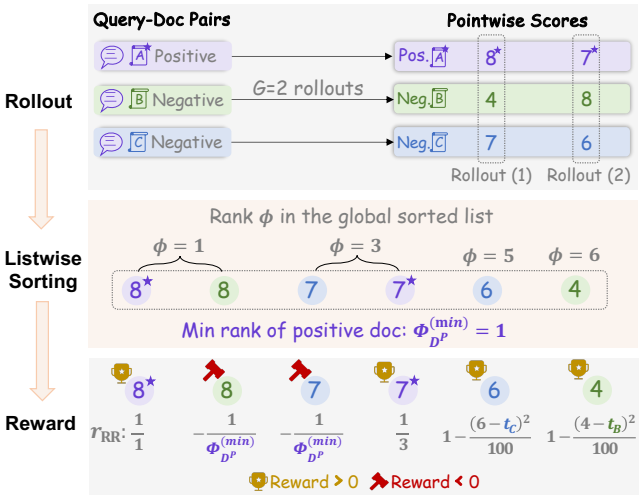

Figure 7: Example for rule-based listwise reward r_{RR} when there are $G = 2$ rollouts and $N = 3$ documents for query q.

clipped importance sampling estimator, which evaluates the advantage of each trajectory relative to others in the group. To prevent the policy from deviating too drastically from the robust SFT model, we incorporate a Kullback Leibler (KL) divergence penalty. This term regularizes the policy updates, ensuring the model learns a more nuanced scoring function without sacrificing its foundational knowledge. The complete objective function is formulated as follows:

$$J_{GRPO}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\{y_i\}\sim\pi_{\theta_{\text{old}}}} \left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|y_i|}\sum_{t=1}^{|y_i|}(\mathcal{C} - \beta D_{\text{KL}}) \right],$$

where the clipped estimator $\mathcal{C}$ and KL penalty D_{KL} are:

$$\mathcal{C} = \min\Big(\frac{\pi_\theta(y_{i,t}|x,y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t}|x,y_{i,<t})}\hat{A}_{i,t}, \text{clip}\left(\frac{\pi_\theta(y_{i,t}|x,y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t}|x,y_{i,<t})}, 1-\epsilon, 1+\epsilon\right)\hat{A}_{i,t}\Big),$$

$$D_{\text{KL}} = \frac{\pi_{\text{ref}}(y_{i,t}|x,y_{i,<t})}{\pi_\theta(y_{i,t}|x,y_{i,<t})} - \log\frac{\pi_{\text{ref}}(y_{i,t}|x,y_{i,<t})}{\pi_\theta(y_{i,t}|x,y_{i,<t})} - 1,$$

where the advantage $\hat{A}_{i,t}$ is computed by normalizing the reward $r(y_i)$ of trajectory y_i using the mean and standard deviation of rewards in the group: $\hat{A}_{i,t} = \frac{r(y_i) - \text{mean}(\{r(y_j)\}_{j=1}^G)}{\text{std}(\{r(y_j)\}_{j=1}^G)}$. The hyperparameters β and ϵ control the strength of the KL penalty and the clipping range, respectively.

Listwise Reranking Reward Design. Previous studies have established that listwise rerankers often outperform pointwise models because they directly optimize the relative ordering of documents. Drawing inspiration from this, we designed a novel rule-based listwise reward function, r_{RR}. While our model produces scores pointwise at inference, this reward design allows it to learn from relative document ordering during training, a key strength of listwise methods.

As shown in Figure 7, for a given query, we first generate scores for all N candidate documents and their corresponding G rollouts. These $N \times G$ scores are then aggregated and sorted to determine the global rank of each generated output. Our reward function, r_{RR}, operates based on the following principles. Positive documents receive a high reward based on their reciprocal rank, incentivizing them to be placed as high as possible. Negative documents that are incorrectly ranked higher than any positive document receive a substantial penalty. Negative documents ranked correctly below all positive documents receive a smooth reward based on the squared error against a reference score from the SFT model, which helps maintain a stable scoring distribution. A significant penalty is assigned if the model's output is not correctly formatted, which discourages generation errors.

The formal definition of the reward r_{RR} is as follows, where $\phi_i^{(j)}$ is the global rank of the j-th rollout for document d_i with ties resolved by assigning the minimum rank. Let $\mathcal{D}^P$ and $\mathcal{D}^N$ be the set of positive and negative documents, respectively. Let $\Phi_{\mathcal{D}^P}^{(\text{min})}$ and $\Phi_{\mathcal{D}^P}^{(\text{max})}$ denote the minimum and maximum ranks for positive documents, respectively.

$$r_{\text{RR}} = \begin{cases} \frac{1}{\phi_i^{(j)}}, & \text{formatted, } d_i \in \mathcal{D}^P, \\ -\frac{1}{\Phi_{\mathcal{D}^P}^{(\text{min})}}, & \text{formatted, } d_i \in \mathcal{D}^N, \phi_i^{(j)} \le \Phi_{\mathcal{D}^P}^{(\text{max})}, \\ 1 - \frac{(s_i - t_i)^2}{100}, & \text{formatted, } d_i \in \mathcal{D}^N, \phi_i^{(j)} > \Phi_{\mathcal{D}^P}^{(\text{max})}, \\ -1, & \text{otherwise,} \end{cases}$$

where "formatted" condition requires that the model's output conforms to the expected structure so a score can be extracted. In the third case, s_i is the generated score and t_i is a reference score from the SFT model π_{ref}.

We trained the model using a randomly sampled subset of $2,048$ queries from the SFT dataset, each paired with 20 documents. The GRPO algorithm was applied directly using these pointwise prompts without any pre-collected trajectories or external reward signals. The SFT tuned model served a dual role as both the initial policy for training and the reference model π_{ref} for calculating KL divergence and providing reference scores for the reward function.

Stage	# Queries	# Doc per query	# Query-doc pairs
SFT	14,799	20	295,980
RL	2,048	20	40,960

Table 2: Statistics of training data.

4 Experiment

4.1 Evaluation Setup

Benchmarks. We evaluate our method across a diverse set of reranking tasks on four benchmark suites. For in-domain semantic matching, we use the TREC DL19 and DL20 passage ranking collections (Craswell et al. 2020, 2021). For out-of-domain generalization, we utilize the entire BEIR benchmark (Thakur et al. 2021), and report results on a five-dataset subset termed BEIR-5 (ArguAna, DBPedia, FiQA, NFCorpus, and SCIDOCS) to facilitate efficient ablation studies. To assess complex reasoning, we employ the BRIGHT benchmark (Su et al. 2024) for general reasoning

Method		Average	Reasoning-intensive Relevance		Semantic Relevance	
			BRIGHT	FollowIR	BEIR	TREC DL
First-stage retriever		25.9	13.7	0	40.8	49.3
Listwise	Rank-R1-7B (Zhuang et al. 2025)	34.6	15.7	3.6	**49.0**	<u>70.0</u>
	Rearank-7B (Zhang et al. 2025)	35.3	17.4	2.3	**49.0**	**72.5**
Pointwise	JudgeRank-8B (Niu et al. 2024)	32.1	17.0	9.9	39.1	62.6
	Rank1-7B (Weller et al. 2025b)	34.6	18.2	9.1	44.2	67.1
	QwQ-32B powered reranker **(Ours)**	<u>37.7</u>	<u>23.2</u>	**14.1**	47.5	66.1
	ERANK-4B **(Ours)**	36.8	22.7	11.0	44.8	68.9
	ERANK-14B **(Ours)**	36.9	23.1	10.3	47.1	67.1
	ERANK-32B **(Ours)**	**38.1**	**24.4**	<u>12.1</u>	<u>47.7</u>	68.1

Table 3: Evaluation on different relevance types using original queries without hybrid scores. FollowIR uses p-MRR as the evaluation metric while the others use nDCG@10. Best results are indicated in bold, while second-best results are underlined.

and the FollowIR benchmark (Weller et al. 2025a) for instruction following abilities.

Baselines. We compare our model, ERANK, against leading reasoning-based rerankers. These include JudgeRank-8B, a zero-shot reranker that uses a multi-step reasoning process; Rank1-7B, a pointwise reranker fine-tuned via distillation; Rank-R1-7B, a listwise reranker trained with the GRPO algorithm; and Rearank-7B, a state-of-the-art listwise model trained to predict optimal document permutations. Additionally, we include two top-performing listwise rerankers from the online BRIGHT benchmark (Su et al. 2025): Rank-R1-32B-v0.2 (ielabgroup 2025) trained on the ReasonIR training set, and the zero-shot XRR-Gemini-2.5-Flash (jataware 2025) which performs a two-pass reranking process.

4.2 Implementation and Evaluation Details

For TREC DL, BEIR, and BRIGHT benchmarks, we rerank the top 100 candidates retrieved by BM25 with Pyserini (Lin et al. 2021). For FollowIR, we rerank the 1,000 candidates provided by the benchmark. Evaluation is performed using nDCG@10 for the first three benchmarks and preference-based Mean Reciprocal Rank (p-MRR) for FollowIR. In all cases, higher scores indicate better performance.

Following prior studies (Weller et al. 2025b; Niu et al. 2024; ielabgroup 2025), we adopt similar settings for a thorough evaluation on the challenging BRIGHT benchmark. First, to improve first-stage retrieval precision, we use reasoning queries expanded by GPT-4. Documents are then retrieved using either BM25 or the ReasonIR-8B model (Shao et al. 2025). During the reranking phase, these expanded queries are not provided to the reranker. Second, we employ a hybrid strategy to combine BM25 scores and reranking scores for low-cost model ensembling. While JudgeRank uses a simple weighted sum and Rank-R1-32B-v0.2 adopts min-max normalization, our ERANK applies standardization before score aggregation, as detailed in Appendix G.

Using Qwen3 LLM series (Qwen Team 2025a) as the backbone for ERANK, the first training stage consists of one epoch of Supervised Fine-Tuning (SFT) with Low-Rank Adaptation (LoRA) (Hu et al. 2022). The second stage uses the GRPO algorithm (Shao et al. 2024) for Reinforcement Learning (RL), performing full-parameter fine-tuning for 10 epochs with a group $G = 5$. Detailed hyperparameters can be found in Appendices E and F. Table 2 presents the number of queries used for SFT and RL training. All experiments are conducted on four NVIDIA A100 (80GB) GPUs. We use official checkpoints for all baselines and reproduce JudgeRank based on its published methodology.

4.3 Main Results

Table 3 presents the main results, with detailed reports available in Appendix H. On average, ERANK-4B clearly outperforms all pointwise baselines with 7B or 8B parameters. Furthermore, ERANK-4B significantly surpasses listwise rerankers, which are typically more powerful, on reasoning-intensive tasks despite having fewer parameters. This demonstrates that ERANK-4B achieves superior effectiveness while maintaining lower latency compared to listwise methods. Beyond the 4B model, we extend our two-stage training pipeline to Qwen3-14B and Qwen3-32B models using identical training data. The results show an overall performance improvement with increased model size, indicating a clear scaling trend. At the 32B scale, our trained ERANK-32B reranker outperforms its teacher model, QwQ-32B, which confirms the efficacy of our training procedure.

Table 4 further reports results with advanced retrieval and BM25 hybrid strategy on the BRIGHT benchmark. Our ERANK rerankers consistently achieve state-of-the-art performance compared to baselines of similar model size, showing superior robustness and effectiveness. Despite using a pointwise paradigm, ERANK-4B achieves a notable nDCG@10 of 38.7 with the BM25 hybrid on documents retrieved by ReasonIR-8B. The ERANK-32B model with BM25 hybrid achieves an nDCG@10 of 40.2, outperforming the state-of-the-art Rank-R1-32B-v0.2 listwise reranker. Moreover, ERANK-32B approaches the performance of XRR2, a listwise method that employs the Gemini-2.5-Flash model.

4.4 Analysis

Impact of training stages. To investigate the contribution of SFT and RL to reranking ability, we perform an ablation study using consistent prompts across different model variants. As shown in Table 5, both SFT and RL independently

Method	nDCG@10
Retrieve by BM25 using GPT-4 reason-query	
BM25	27.0
Rank-R1-7B (Zhuang et al. 2025)	23.9
Rank1-7B (Weller et al. 2025b)	25.5
Rearank-7B (Zhang et al. 2025)	29.1
XRR2-Gemini-2.5-Flash (jataware 2025)	**40.3***
JudgeRank-8B (Niu et al. 2024)	24.4
+ *BM25 hybrid*	31.0
ERANK-4B **(Ours)**	32.9
+ *BM25 hybrid*	36.1
ERANK-14B **(Ours)**	33.5
+ *BM25 hybrid*	36.7
ERANK-32B **(Ours)**	34.6
+ *BM25 hybrid*	37.4
Retrieve by ReasonIR-8B using GPT-4 reason-query	
ReasonIR-8B (Shao et al. 2025)	30.5
Rank-R1-7B (Zhuang et al. 2025)	24.1
Rank1-7B (Weller et al. 2025b)	24.3
Rearank-7B (Zhang et al. 2025)	27.5
JudgeRank-8B (Niu et al. 2024)	20.2
+ *BM25 hybrid*	22.7
Rank-R1-32B-v0.2 (ielabgroup 2025)	37.7*
+ *BM25 hybrid*	40.0*
ERANK-4B **(Ours)**	30.5
+ *BM25 hybrid*	38.7
ERANK-14B **(Ours)**	31.8
+ *BM25 hybrid*	39.3
ERANK-32B **(Ours)**	32.8
+ *BM25 hybrid*	**40.2**

Table 4: Further evaluation on BRIGHT benchmark. *Taken from BRIGHT online website (Su et al. 2025).

yield significant improvements, and solely applying RL still achieves substantial gains without any teacher model. Our two-stage training pipeline for ERANK-4B yields the most robust effectiveness overall.

Varying rewards in RL training. Besides the rule-based listwise reward r_{RR} using Reciprocal Rank, we evaluate two different rule-based rewards, which are briefly described as follows. Please refer to Appendix I for more details.

(1) Pointwise reward r_{SE}. It uses squared error to measure the difference between the score s_i from the policy model and the score t_i from the teacher model (i.e., QwQ-32B).

(2) Listwise reward r_{nDCG}. This is a listwise reward similar to r_{RR}, which assesses how effectively positive documents contribute to the nDCG metric while penalizing negative documents ranked above any positive document.

Table 6 compares the results when utilizing different rewards for GRPO training. Overall, listwise rewards such as r_{nDCG} and r_{RR} lead to better outcomes than the pointwise reward r_{SE}. Pointwise reward that mimicks the teacher model's scores for each document independently may not align well with global ranking objectives. In contrast, listwise rewards tend to yield more favorable results by considering relative ranks to encourage a better final reranking order. While r_{nDCG} shows a notable improvement on the FollowIR benchmark, r_{RR} demonstrates greater robustness and superior overall performance across the four benchmarks.

	Avg.	BRIGHT	FollowIR	BEIR-5	TREC DL
Qwen3-4B	12.7	3.6	1.9	6.4	39.0
SFT Only	32.8	22.0	11.2	30.0	68.1
RL Only	31.8	20.1	**12.2**	28.2	66.5
SFT+RL	**33.8**	**22.7**	11.0	**32.4**	**68.9**

Table 5: Performance of different training stages.

	Avg.	BRIGHT	FollowIR	BEIR-5	TREC DL
Before RL	32.8	22.0	11.2	30.0	68.1
r_{SE}	31.1	22.3	8.7	30.8	62.6
r_{nDCG}	**33.8**	21.5	**13.2**	**32.8**	67.6
r_{RR}	**33.8**	**22.7**	11.0	32.4	**68.9**

Table 6: Performance of different rewards for RL training.

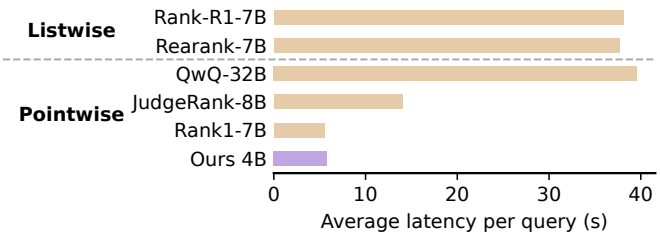

Figure 8: Latency for returning the complete reranked list per query, averaged on all queries of TREC DL19 dataset.

Reranking latency. In real-world applications, achieving superior performance across diverse relevance types must be balanced with acceptable latency. As shown in Figure 8, pointwise methods offer significantly lower latency than their listwise counterparts. This advantage stems from the ability of pointwise methods to process documents in parallel, whereas listwise methods require sequential processing as discussed in Section 2. Specifically, the ERANK-4B reranker is six times faster than both the listwise methods and the QwQ-32B pointwise reranker, highlighting its practicality for real-world applications. Compared to Rank1-7B, ERANK-4B achieves superior performance by generating more tokens while maintaining comparable latency.

5 Conclusion

In this paper, we introduce ERANK, an LLM-based reranker designed for effective and efficient reranking of documents in both semantic and reasoning-intensive tasks. To support real-world applications, ERANK adopts the pointwise paradigm to ensure low latency while achieving competitive performance through a two-stage training pipeline. The first stage conducts Supervised Fine-Tuning (SFT) to build foundational reasoning capabilities, and the second stage employs the GRPO algorithm with a novel rule-based listwise reward tailored for pointwise rerankers. Extensive evaluation on four benchmarks demonstrates the effectiveness and robustness of ERANK compared to state-of-the-art methods.

Acknowledgments

This work was substantially supported by National NSF of China (62572126) and Key Projects of the National NSF of China (U23A20496).

References

Bajaj, P.; Campos, D.; Craswell, N.; Deng, L.; Gao, J.; Liu, X.; Majumder, R.; McNamara, A.; Mitra, B.; Nguyen, T.; et al. 2016. MS MARCO: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

Chuang, Y.-N.; Chen, C.-M.; Wang, C.-J.; Tsai, M.-F.; Fang, Y.; and Lim, E.-P. 2020. TPR: Text-aware preference ranking for recommender systems. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 215–224.

Craswell, N.; Mitra, B.; Yilmaz, E.; and Campos, D. 2021. Overview of the TREC 2020 deep learning track. arXiv:2102.07662.

Craswell, N.; Mitra, B.; Yilmaz, E.; Campos, D.; and Voorhees, E. M. 2020. Overview of the TREC 2019 deep learning track. arXiv:2003.07820.

DeepSeek AI. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv:2501.12948.

Gao, J.; Chen, B.; Zhao, X.; Liu, W.; Li, X.; Wang, Y.; Wang, W.; Guo, H.; and Tang, R. 2025. Llm4rerank: Llm-based auto-reranking framework for recommendations. In *Proceedings of the ACM on Web Conference 2025*, 228–239.

Gao, L.; Dai, Z.; and Callan, J. 2021. Rethink training of BERT rerankers in multi-stage retrieval pipeline. In *European Conference on Information Retrieval*, 280–286. Springer.

Gupta, S.; Ranjan, R.; and Singh, S. N. 2024. A comprehensive survey of retrieval-augmented generation (RAG): Evolution, current landscape and future directions. *arXiv preprint arXiv:2410.12837*.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W.; et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2): 3.

Huang, X.; Liu, W.; Chen, X.; Wang, X.; Wang, H.; Lian, D.; Wang, Y.; Tang, R.; and Chen, E. 2024. Understanding the planning of LLM agents: A survey. *arXiv preprint arXiv:2402.02716*.

ielabgroup. 2025. Rank-R1-32B-v0.2. https://huggingface.co/ielabgroup/Rank-R1-32B-v0.2. Accessed: 2025-07-24.

jataware. 2025. XRR2: Expand $\rightarrow$ Retrieve $\rightarrow$ Rerank $\rightarrow$ Rerank - simple method with strong results on BRIGHT benchmark. https://github.com/jataware/XRR2. Accessed: 2025-07-24.

Lee, J.; Yun, S.; Kim, H.; Ko, M.; and Kang, J. 2018. Ranking Paragraphs for Improving Answer Recall in Open-Domain Question Answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 565–569.

Li, X.; Wang, S.; Zeng, S.; Wu, Y.; and Yang, Y. 2024. A survey on LLM-based multi-agent systems: workflow, infrastructure, and challenges. *Vicinagearth*, 1(1): 9.

Liang, P.; Bommasani, R.; Lee, T.; Tsipras, D.; Soylu, D.; Yasunaga, M.; Zhang, Y.; Narayanan, D.; Wu, Y.; Kumar, A.; et al. 2022. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*.

Lin, J.; Ma, X.; Lin, S.-C.; Yang, J.-H.; Pradeep, R.; and Nogueira, R. 2021. Pyserini: A Python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2356–2362.

Lin, J.; Nogueira, R.; and Yates, A. 2022. *Pretrained transformers for text ranking: Bert and beyond*. Springer Nature.

Liu, J.; Ma, Y.; Zhao, R.; Zheng, J.; Ma, Q.; and Kang, Y. 2025. ListConRanker: A Contrastive Text Reranker with Listwise Encoding. *arXiv preprint arXiv:2501.07111*.

Liu, Z.; Zhou, Y.; Zhu, Y.; Lian, J.; Li, C.; Dou, Z.; Lian, D.; and Nie, J.-Y. 2024. Information retrieval meets large language models. In *Companion Proceedings of the ACM Web Conference 2024*, 1586–1589.

Ma, X.; Wang, L.; Yang, N.; Wei, F.; and Lin, J. 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*, 2421–2425.

Ma, X.; Zhang, X.; Pradeep, R.; and Lin, J. 2023. Zero-shot listwise document reranking with a large language model. *arXiv preprint arXiv:2305.02156*.

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

OpenAI. 2024. OpenAI o1 System Card. arXiv:2412.16720.

Qin, Z.; Jagerman, R.; Hui, K.; Zhuang, H.; Wu, J.; Yan, L.; Shen, J.; Liu, T.; Liu, J.; Metzler, D.; et al. 2024. Large Language Models are Effective Text Rankers with Pairwise Ranking Prompting. In *Findings of the Association for Computational Linguistics: NAACL 2024*, 1504–1518.

Qwen Team. 2025a. Qwen3 Technical Report. arXiv:2505.09388.

Qwen Team. 2025b. QwQ-32B: Embracing the Power of Reinforcement Learning.

Shao, R.; Qiao, R.; Kishore, V.; Muennighoff, N.; Lin, X. V.; Rus, D.; Low, B. K. H.; Min, S.; Yih, W.-t.; Koh, P. W.; et al. 2025. ReasonIR: Training Retrievers for Reasoning Tasks. *arXiv preprint arXiv:2504.20595*.

Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y.; Wu, 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.-y.; Liu, H.; Shi, Q.; Siegel, Z. S.; Tang, M.; Sun, R.; Yoon, J.; Arik, S. O.; Chen, D.; and Yu, T. 2024. BRIGHT: A Realistic and Challenging Benchmark for Reasoning-Intensive Retrieval.

Su, H.; Yen, H.; Xia, M.; Shi, W.; Muennighoff, N.; Wang, H.-y.; Liu, H.; Shi, Q.; Siegel, Z. S.; Tang, M.; Sun, R.; Yoon, J.; Arik, S. O.; Chen, D.; and Yu, T. 2025. BRIGHT Benchmark Online Website. https://brightbenchmark.github.io/. Accessed: August 26, 2025.

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. *arXiv preprint arXiv:2304.09542*.

Thakur, N.; Reimers, N.; Rücklé, A.; Srivastava, A.; and Gurevych, I. 2021. BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Wang, X.; Wang, Z.; Gao, X.; Zhang, F.; Wu, Y.; Xu, Z.; Shi, T.; Wang, Z.; Li, S.; Qian, Q.; et al. 2024. Searching for best practices in retrieval-augmented generation. *arXiv preprint arXiv:2407.01219*.

Weller, O.; Chang, B.; MacAvaney, S.; Lo, K.; Cohan, A.; Van Durme, B.; Lawrie, D.; 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 (Volume 1: Long Papers)*, 11926–11942. Albuquerque, New Mexico: 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*.

Weller, O.; Van Durme, B.; Lawrie, D.; Paranjape, A.; Zhang, Y.; and Hessel, J. 2024. Promptriever: Instruction-trained retrievers can be prompted like language models. *arXiv preprint arXiv:2409.11136*.

Wu, S.; Xiong, Y.; Cui, Y.; Wu, H.; Chen, C.; Yuan, Y.; Huang, L.; Liu, X.; Kuo, T.-W.; Guan, N.; et al. 2024. Retrieval-augmented generation for natural language processing: A survey. *arXiv preprint arXiv:2407.13193*.

Zhang, J.; Chen, Y.; Liu, C.; Niu, N.; and Wang, Y. 2023. Empirical evaluation of ChatGPT on requirements information retrieval under zero-shot setting. In *2023 International Conference on Intelligent Computing and Next Generation Networks (ICNGN)*, 1–6. IEEE.

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

Zhang, L.; Zhang, Y.; Long, D.; Xie, P.; Zhang, M.; and Zhang, M. 2024. A Two-Stage Adaptation of Large Language Models for Text Ranking. In *ACL (Findings)*.

Zhang, Y.; Long, D.; Xu, G.; and Xie, P. 2022. HLATR: enhance multi-stage text retrieval with hybrid list aware transformer reranking. *arXiv preprint arXiv:2205.10569*.

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 *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 38–47.