

Retrieval-GRPO: Reinforcement Learning for Dense Retrieval in Taobao Search[†]

Xingxian Liu*, Dongshuai Li, Jiahui Wan, Tao Wen, Gui Ling,
Fuyu Lv, Yuliang Yan, Dan Ou, Haihong Tang, Bo Zheng

Taobao & Tmall Group of Alibaba
liuxingxian.lxx@alibaba-inc.com

Abstract

Dense retrieval is a fundamental component of e-commerce search, projecting user queries and items into a shared semantic representation space for large-scale, real-time retrieval. While LLM-based embedding models are increasingly replacing BERT-style encoders, prevalent training pipelines still largely rely on BERT-like supervised fine-tuning and hard negative mining. Such paradigms require intricate offline negative construction, thereby constraining model iteration efficiency and limiting the evolution of learned semantic representations. We propose Retrieval-GRPO, a dense retrieval framework based on reinforcement learning. It dispenses with offline hard negative mining by performing dynamic Top-K candidates retrieval during training. Moreover, we introduce a relevance-oriented LLM as a reward model to provide real-time supervisory signals. Guided by the resulting reinforcement learning objective, the retriever updates embeddings to better capture semantic correspondence and achieve improved generalization on complex long-tail queries. Extensive offline and online experiments substantiate the effectiveness of Retrieval-GRPO in Taobao Search, which serves China’s largest e-commerce platform.

1 Introduction

Dense retrieval (Guo et al., 2016, 2020; Mitra and Craswell, 2017) has emerged as a critical technology in e-commerce search engines to address the unique challenge of precisely aligning massive items with user queries. This approach employs advanced pre-trained embedding models to encode user queries and item descriptions into a unified semantic vector space, enabling real-time semantic search over billions of items via Approximate Nearest Neighbor (ANN) algorithms (Malkov

and Yashunin, 2018). The methodology effectively learns query-item alignment relationships within a unified semantic space, even in the absence of direct lexical overlap between them. Dense retrieval directly improves users’ search experience and merchant exposure efficiency, thereby driving significant improvements in overall platform transaction conversion rates.

Previous approaches primarily relied on encoder-only architectures like BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019), utilizing the bidirectional attention mechanism to produce contextual semantic embeddings. Benefiting from rapid advancements in large language model technologies, emerging approaches such as Gemini Embedding (Lee et al., 2025b), Seed Embedding (ByteDance Seed, 2025), QWEN3-Embedding (Zhang et al., 2025), and NV-Embed (Lee et al., 2025a) employ LLMs as encoders to generate semantic embeddings for retrieval, leveraging their extensive knowledge bases and zero-shot generalization capabilities. Despite incorporating LLMs, most current methods still adopt training paradigms designed for BERT-like models, enhancing performance on long-tail challenging queries through continued supervised fine-tuning (SFT) and persistent hard negative sample mining. Notably, offline hard negative sample mining constitutes a complex systemic workflow, which restricts iteration speed and limits the evolution potential of representation capabilities.

In this paper, we propose Retrieval-GRPO, a reinforcement learning method for dense retrieval. This approach eliminates the need for offline manual construction of hard negative samples. Instead, during training, we dynamically retrieve top-k items for each query and employ a relevance LLM (Dong et al., 2025) as the reward model to generate scores. These relevance scores serve as reward signals, enabling real-time user preference alignment and error correction through RL training.

*Corresponding author.

[†]An earlier version of this work appeared as arXiv:2511.13885.

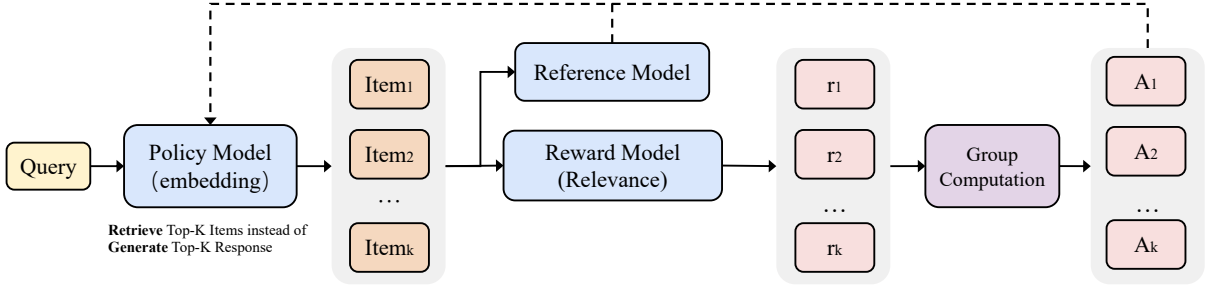


Figure 1: R-GRPO consists of 3 steps: (1)Candidate Selection; (2)Reward Computation; (3)Policy Optimization

In summary, compared to conventional dense retrieval paradigms, Retrieval-GRPO offers three key contributions:

- Eliminates laborious offline hard negative sample mining, resolving protracted iteration cycles between sample selection and training.
- Utilizes candidates sampled by the latest gradient-updating model during training, whereas offline hard negatives derive from static older models. This enables instant learning from relevance model feedback for real-time error correction.
- Conducts comprehensive experiments in Taobao Search, demonstrating the effectiveness of reinforcement learning in dense retrieval with significant improvements in both offline metrics and online A/B testing.

2 Methods

Given a query q_i and an item collection $\mathcal{D} = \{d_1, \dots, d_n\}$, dense retrieval employs a dual-encoder architecture (Reimers and Gurevych, 2019) to independently encode them into fixed-dimensional vectors $\mathbf{q}_i = f_\theta(q_i)$ and $\mathbf{d}_j = f_\theta(d_j)$, then retrieves the top-k items by semantic similarity (e.g., cosine similarity) between $\mathbf{q}_i$ and $\mathbf{d}_j$.

Our training framework consists of two stages. First, SFT is performed to endow the model with discriminative capabilities on simple positive and negative samples. Then, Retrieval-GRPO leverages reinforcement learning with real-time relevance feedback for error correction, eliminating the need for traditional offline construction of hard negative samples (Li et al., 2024).

2.1 SFT

Traditionally, in the SFT phase of dense retrieval, training samples consist of $\langle \text{query}, \text{item} \rangle$ pairs for positive samples, while negative samples are not specifically collected but obtained through in-batch negative sampling (Radford et al., 2021). The InfoNCE loss pulls positive query-item pairs closer while pushing apart irrelevant ones in the embedding space:

$$\mathcal{L}_{\text{InfoNCE}} = -\log \frac{\exp(s(q_i, d_j^+)/\tau)}{\sum_{d_j \in \mathcal{B}} \exp(s(q_i, d_j)/\tau)} \quad (1)$$

where $s(\cdot)$ denotes similarity scoring, τ is the temperature parameter, and $\mathcal{B}$ is the set of in-batch negatives.

However, this approach leads to only those items present as positive samples in the training set being utilized as negative samples for other queries. Meanwhile, items in the product pool that lack historical user interactions are entirely ignored during training. To address this, we enhance the conventional InfoNCE in-batch negative sampling by introducing global negative sampling, which randomly selects items from the product pool to serve as negative samples for queries. Consequently, the negative samples are derived partially from in-batch sampling and partially from the product pool, ensuring that all items in the product pool are exposed to the model during training. The loss is as follows:

$$\mathcal{L} = -\log \frac{\exp(s(q_i, d_j^+)/\tau)}{\underbrace{\sum_{d_j \in \mathcal{B}} \exp(s(q_i, d_j)/\tau)}_{\text{pos and in-batch neg}} + \underbrace{\sum_{d_k^- \in \mathcal{G}} \exp(s(q_i, d_k^-)/\tau)}_{\text{global neg}}} \quad (2)$$

where $\mathcal{G}$ represents global negatives randomly sampled from the global item-pool. This dual-sampling strategy ensures all items in the pool par-

ticipate in training, with negatives comprising both in-batch $\mathcal{B}$ and global $\mathcal{G}$ query-item pairs.

2.2 Retrieval-GRPO

After SFT, the model can distinguish simple positive and negative samples. Conventionally, hard negative mining is introduced to further improve performance on long-tail queries, but this requires complex offline construction pipelines that hinder iteration efficiency.

We propose Retrieval-GRPO, which frames dense retrieval as a reinforcement learning problem to eliminate offline hard negative mining. We model item retrieval as a single-step decision process: the retrieval model serves as the *policy*, selecting items from a candidate set is the *action*, and relevance scores from an LLM judge provide the *reward*. The policy is optimized to increase selection probability for high-reward items and suppress low-reward ones, with the reward magnitude continuously modulating the update strength.

As illustrated in Figure 1, the training pipeline consists of three steps:

- **Candidate Selection:** For each query, the current retrieval model selects top-k items from a large cross-device gathered batch as candidates.
- **Reward Computation:** An LLM-based relevance model assigns continuous relevance scores to each query-candidate pair, serving as the reward signal.
- **Policy Optimization:** Group-relative advantages are computed via z-score normalization of rewards, and the retrieval model is optimized using a reward-weighted policy gradient objective (Equation 5).

2.2.1 Candidate Selection

In the Retrieval-GRPO framework, instead of offline hard negative samples construction, we treat the top-k items retrieved by the dense retrieval model as candidate predictions requiring correction. However, identifying top-k items for every query across the entire product pool is computationally intensive. We simplify this process by gathering items across devices to form a large batch $\hat{B}$ (where $k \ll \hat{B}$), then selecting the top-k candidates from this batch for reward computation. Since retrieving top-k from the full item pool with the continuously updating model at every training step is computationally prohibitive, the cross-device gathered batch serves as a practical approximation. The

ratio $k/|\hat{B}|$ is set slightly above the online retrieval quota-to-pool ratio, providing a marginally denser candidate distribution during training that encourages finer-grained discrimination among borderline candidates.

2.2.2 Relevance Reward

The query-item relevance score is computed by TaoSR1 (Dong et al., 2025), a 42B-MoE LLM trained through multi-stage RL learning. TaoSR1 is fully frozen during training and invoked once per candidate item per training step. It assigns continuous relevance scores to each query-candidate pair, serving as the reward signal for the policy optimization.

2.2.3 GRPO Loss

We formulate dense retrieval as a single-step item selection problem. Given a query q and its top-k candidate group $\{d_1, \dots, d_G\}$, we define the item selection probability as:

$$\pi_\theta(d_i|q) = \frac{\exp(s(q, d_i)/\tau)}{\sum_{j=1}^G \exp(s(q, d_j)/\tau)} \quad (3)$$

where $s(q, d_i)$ is the cosine similarity between the query and item embeddings produced by the retrieval model, and τ is a temperature parameter.

The relevance reward model assigns a scalar reward r_i to each candidate. We compute the group-relative advantage via z-score normalization:

$$\hat{A}_i = \frac{r_i - \text{mean}(\{r_j\}_{j=1}^G)}{\text{std}(\{r_j\}_{j=1}^G)} \quad (4)$$

The advantage $\hat{A}_i$ captures relative relevance preferences within the group: items with above-average rewards receive positive advantages and are pulled closer to the query, while those with below-average rewards receive negative advantages and are pushed away, with the magnitude continuously modulated by the reward signal.

Unlike standard GRPO in text generation, where clipping is required because multiple gradient steps are taken on the same set of sampled responses, our retrieval setting re-samples candidates on-policy at each training step. The importance sampling ratio thus remains close to 1, making clipping unnecessary. This on-policy simplification connects R-GRPO to the REINFORCE algorithm with a group-relative baseline, while retaining the KL-regularized framework of GRPO. The final objec-

tive simplifies to:

$$\mathcal{J}(\theta) = \mathbb{E} \left[\frac{1}{G} \sum_{i=1}^G \hat{A}_i \log \pi_{\theta}(d_i | q) \right] - \beta \mathbb{D}_{\text{KL}}[\pi_{\theta} \| \pi_{\text{ref}}]. \quad (5)$$

where π_{ref} is the distribution of the SFT-stage model, serving as a reference to prevent excessive deviation of the learned embeddings. The KL regularization term preserves the semantic structure acquired during supervised fine-tuning, while allowing the reinforcement learning objective to refine representations for challenging cases.

Distinction from Knowledge Distillation. Although Retrieval-GRPO uses a large frozen LLM as the reward model, it differs fundamentally from Knowledge Distillation (KD). In KD, the student directly mimics the teacher’s output distribution via soft labels and KL divergence minimization. In contrast, the reward model in R-GRPO provides only scalar relevance scores and does not expose its output distribution to the retriever. The retriever explores candidates on-policy and updates its representations through advantage-weighted policy gradients, where the reward model serves as a *judge* rather than a *teacher*.

3 Experiment

3.1 Experiment Setup

Full implementation details are provided in Appendix B.

3.1.1 Dataset and Metrics

We construct three offline test sets from the platform’s search logs: (1) **General Retrieval** (50k queries, 2M pairs), (2) **Long-Tail Query** (10k queries, 300k pairs), and (3) **Long-Tail Experience** (8k queries covering negation, substitutes, QA, and knowledge-based queries, identified as the most challenging grammatical patterns through analysis of search logs). All test sets are independently annotated by human annotators; TaoSR1 is not used in any evaluation stage. We report two offline metrics: **Hitrate@6k**, the proportion of ground-truth items in the top-6k retrieved results. The cutoff is the production quota of the semantic retrieval channel: the deployed system retrieves from a quality-filtered pool of roughly 10^8 items and passes exactly 6,000 candidates per query to downstream ranking. It is therefore the system’s operating point rather than an arbitrary evaluation

cutoff. Although training selects top-8 candidates from a 32,768-item cross-device pool, both settings evaluate the highest-scoring tail of a much larger candidate set; the training value $k = 8$ additionally balances hard-example coverage against stable within-group normalization. **Goodrate@100** is the proportion of items in the top-100 results rated as relevant, where relevance is annotated on a four-level scale: L1 (Irrelevant), L2 (Marginal), L3 (Related), and L4 (Excellent).

For online evaluation, we conduct A/B testing with human evaluators assessing three metrics: **GSB** (pairwise win rate over the baseline), **Query Goodrate** (percentage of queries with good or mid page-level relevance), and **Item Goodrate** (average proportion of highly relevant items per request).

3.1.2 Baselines

- BERT-base: A 12-layer 110M BERT trained via the RetroMAE and contrastive learning.
- Query-Rewrite: Query rewriting via CSA-QR (Feng et al., 2025), followed by retrieval via inverted index.
- Qwen-3B (Yang et al., 2025): Dense retrieval model trained based on Qwen-3B, modified to use bi-directional attention.
- Qwen3-VL-Embedding-2B + SFT: A state-of-the-art general embedding backbone of comparable size, adapted using the same in-domain SFT data and protocol as our backbone.
- Tbstars Post-Pretrain(base): Dense retrieval model trained based on an internal closed-source Tbstars with 3B parameters, modified to use bi-directional attention.
- Tbstars SFT + Ranking loss(Hard Neg): After post-pretrain on Tbstars, performing SFT and pairwise ranking training on Tbstars with hard negative samples.
- Tbstars SFT + DPO(Hard Neg): After post-pretrain on Tbstars, performing SFT and DPO training with hard negative samples.
- Tbstars SFT + Logits KD: After SFT on Tbstars, performing logits-based knowledge distillation using TaoSR1 as the teacher model.

3.2 Offline Evaluation

Offline evaluation is conducted using Hitrate@6k and Goodrate@100 metrics. Table 1 presents the offline experience evaluation results from the view of relevance, and Table 2 shows the offline evaluation results from the view of recall.

As shown in Table 1, the model with SFT and

Model	Goodrate@100				
	Q&A	Alternative	Negative	Knowledge	Overall
BERT-base	69.60	26.86	49.90	50.49	49.21
Query-Rewrite	84.52	36.39	57.40	62.65	60.24
Qwen-3B	82.03	34.65	59.87	58.09	58.66
Tbstars Post-Pretrain(base)	80.67	32.87	60.26	63.20	59.25
Tbstars SFT Only	84.01	38.70	62.60	68.45	63.44
Tbstars SFT + Ranking loss(Hard Neg)	84.90	40.12	62.40	68.71	64.03
Tbstars SFT + DPO(Hard Neg)	84.96	40.21	63.60	68.76	64.38
Tbstars SFT + R-GRPO	85.05	43.32	65.80	69.50	65.91

Table 1: Offline Performance Comparison of Training Strategies with Goodrate@100

Model	Hitrate@6k	
	General	Long-Tail
BERT-base	32.28	19.10
Qwen-3B	45.08	33.67
Qwen3-VL-Embedding-2B + SFT	46.65	36.81
Tbstars Post-Pretrain(base)	46.76	32.74
SFT Only	48.00	37.52
R-GRPO Only	46.20	37.79
SFT + Ranking loss(Hard Neg)	48.06	38.11
SFT + DPO(Hard Neg)	48.20	39.35
SFT + Logits KD	48.52	40.02
SFT + R-GRPO	49.35	41.07

Table 2: Offline Performance Comparison of Training Strategies with Hitrate@6k

Retrieval-GRPO achieves optimal performance, outperforming the state-of-the-art baseline Tbstars (SFT only) with an average improvement of 2.47pt across four challenging query slices. The most significant enhancement is observed in the alternative queries, showing a 4.62 pt gain. It is because the alternatives typically share identical functionality with the original items but differ in brand and price. From the lexical overlap metric perspective, queries exhibit higher overlap scores with original brands than their alternatives, posing challenges for models relying primarily on lexical matching. For instance, a search for "alternative PS5 controller" might incorrectly prioritize official Sony products due to brand name overlap, despite the user's explicit request for alternatives. Through Retrieval-GRPO, wrong predictions can be corrected via relevance rewards, which helps the model overcome superficial lexical matches and develop deeper semantic understanding.

As shown in Table 2, the two-stage SFT+Retrieval-GRPO model achieves the best performance, outperforming the current

state-of-the-art baseline by 1.35pt and 3.55pt on general queries and long-tail queries, respectively.

3.3 Effect of R-GRPO

Table 2 shows that SFT+Retrieval-GRPO outperforms SFT alone, offline ranking and DPO, and logits-based KD. To isolate the source of the gain, Table 4 holds the backbone, candidate pool, and TaoSR1 supervision fixed. Replacing static hard negatives with candidates retrieved by the current policy improves DPO by 0.40/0.51 points on general/long-tail queries. Replacing on-policy DPO with R-GRPO adds a further 0.75/1.21 points. Thus both fresher candidates and the group-relative objective contribute.

DPO thresholds TaoSR1 scores into L3/L4 positive and L1/L2 negative pairs. Candidates sharing a binary label have no relative ordering, and an all-positive or all-negative group provides no pairwise signal. R-GRPO instead uses every continuous score through a z-normalized advantage, preserving within-label ordering and producing gradients for every non-degenerate group. The label-source effect is also controlled by logits KD in Table 2, which uses the same frozen TaoSR1 yet trails R-GRPO by 0.83/1.05 points. This evidence attributes the remaining gain to the objective together with the finer-grained signal it can represent.

Removing first-stage SFT and applying R-GRPO directly also degrades performance, showing that basic discrimination must be learned before on-policy correction. The two stages therefore form a progressive-difficulty curriculum.

Figure 3 shows how the distribution of relevance levels for the top-k items evolves during training, where L1 (Irrelevant) and L2 (Marginal) represent low relevance, while L3 (Related) and L4 (Excellent) represent high relevance. As training pro-



Figure 2: Top-k items in R-GRPO and relevance scores assigned by the relevance model for query-item pairs

Query Type	Case	GSB	Query Goodrate	Item Goodrate
Q&A	What medicine can make hair black?	+24.30%	+10.64pt	+8.2pt
Alternative	Alternative PS5 controller	+28.57%	+10.87pt	+9.1pt
Negative	Short sleeves that don't stick to hair	+41.16%	+17.07pt	+11.65pt
Knowledge	Paint that is not afraid of car pressure	+25.5%	+12.65pt	+7.60pt
Overall	-	+29.88%	+12.81pt	+9.14pt

Table 3: Online Side-by-Side Human Evaluations

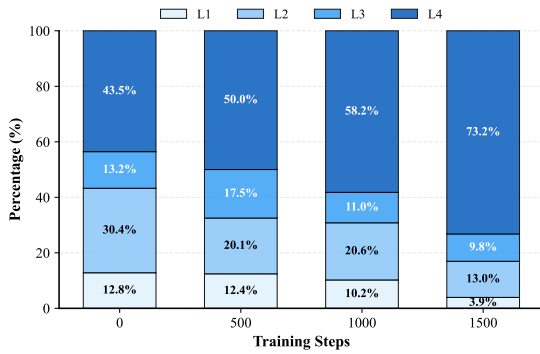


Figure 3: Distribution of Relevance Levels in R-GRPO

Setting	General	Long-Tail
Offline DPO	48.20	39.35
On-policy DPO	48.60	39.86
On-policy R-GRPO	49.35	41.07

Table 4: Factor decomposition on Hitrate@6k. The last two rows use identical on-policy candidates and TaoSR1 supervision.

gresses, L1 and L2 gradually shift towards L3 and L4, indicating that the model learns to retrieve increasingly relevant items.

We also conduct ablation on the reward model by replacing TaoSR1 with smaller relevance models; results are provided in Appendix C.

3.4 Examples in R-GRPO

During R-GRPO training, the model dynamically identifies challenging candidates and adjusts its

representations based on real-time relevance feedback. Figure 2 illustrates the top-k items selected by the dense retrieval model during R-GRPO training, along with the relevance scores assigned by the relevance model to these query-item pairs. As demonstrated in the figure, for the user search query "square stainless steel water tank", among the top-k items, two stainless steel square sinks get high relevance scores (0.99 and 0.99), while a plastic product (0.44) and a sink storage rack (0.29) obtain significantly lower relevance scores.

Additional case studies comparing our model with the online baseline across four types of long-tail queries are provided in Appendix D.

3.5 Online Evaluation

We deploy the model into Taobao Search and validate its effectiveness through online A/B testing. A side-by-side evaluation is conducted between two experimental buckets, where the top 10 items for 2000 actual online queries across four challenging query slices are manually annotated using three human evaluation metrics introduced in subsection 3.1.

As shown in Table 3, the most significant improvement is observed in negative-class queries, as negative semantics and literal overlap metrics typically exhibit opposing trends, posing additional challenges for models that rely on literal matching rather than deep semantic understanding. Through

Retrieval-GRPO, the model leverages real-time relevance feedback to correct wrong predictions during training, significantly improving search experience for queries requiring deep understanding of user intent.

All improvements in Table 3 are statistically significant ($p < 0.05$). Beyond relevance improvements, the deployed model also yields measurable business impact: on long-tail queries, order volume increases by 1.68% and GMV increases by 1.62% in online A/B testing. Note that Retrieval-GRPO introduces additional computation only during training; at deployment, the system is identical to a standard dual-encoder retrieval model with no additional online latency. The reward model is served on a dedicated inference platform, separate from the 128 training GPUs. Our previous offline hard-negative pipeline also performed ANN retrieval and TaoSR1 scoring, so reward inference is not a new cost unique to R-GRPO. Eliminating offline data engineering, index refreshes, and waiting cycles reduces end-to-end iteration wall-clock time to roughly one third of the previous pipeline in our production environment. This relative wall-clock figure includes system orchestration and is not intended as a hardware-independent GPU-hour comparison.

4 Conclusion

We present Retrieval-GRPO, a reinforcement learning framework for dense retrieval that replaces offline hard negative mining with dynamic candidate retrieval and LLM-based relevance rewards. Extensive experiments on a large-scale e-commerce search platform demonstrate significant improvements on both offline metrics and online A/B testing, particularly for complex long-tail queries. Deployed in Taobao Search, serving China’s largest e-commerce platform, Retrieval-GRPO validates the effectiveness of reinforcement learning in improving dense retrieval for real-world search systems.

Limitations

Our work has several limitations. First, Retrieval-GRPO relies on a high-quality reward model; as shown in Appendix C, a weak reward model can degrade performance, indicating that the framework’s effectiveness is contingent on accurate relevance scoring. Second, the two-stage training pipeline requires the SFT stage as a prerequisite (Table 2), and the reward model inference during training

introduces additional computational overhead compared to standard contrastive learning. Third, our experiments use one 3B backbone, proprietary Chinese e-commerce data from Taobao Search, and no public benchmark. Although the method itself is language-agnostic, its generalizability across backbone families, model sizes, languages, and domains remains to be validated. Finally, at deployment time, the framework introduces no additional online latency since only the dual-encoder retrieval model is served, but the training cost on large-scale GPU clusters may limit accessibility for smaller organizations.

References

- ByteDance Seed. 2025. [Seed1.5-Embedding](#). Project page. Accessed 2026-08-28.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.
- Chenhe Dong, Shaowei Yao, Pengkun Jiao, Jianhui Yang, Yiming Jin, Zerui Huang, Xiaojiang Zhou, Dan Ou, Haihong Tang, and Bo Zheng. 2025. [TaoSR1: The thinking model for e-commerce relevance search](#). *arXiv preprint arXiv:2508.12365*.
- Yunling Feng, Gui Ling, Yue Jiang, Jianfeng Huang, Dan Ou, Qingwen Liu, Fuyu Lv, and Yajing Xu. 2025. Complicated semantic alignment for long-tail query rewriting in taobao search based on large language model. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 4435–4446.
- Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. [SimCSE: Simple contrastive learning of sentence embeddings](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910. Association for Computational Linguistics.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, and Xiao Bi. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.
- Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. 2016. [A deep relevance matching model for ad-hoc retrieval](#). In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*.

- Jiafeng Guo, Yixing Fan, Liang Pang, Liu Yang, Qingyao Ai, Hamed Zamani, Chen Wu, W. Bruce Croft, and Xueqi Cheng. 2020. [A deep look into neural ranking models for information retrieval](#). In *Information Processing & Management*, page 102067.
- Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, and Prateek Jain. 2022. Matryoshka representation learning. *Advances in Neural Information Processing Systems*, 35:30233–30249.
- Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2025a. [NV-Embed: Improved techniques for training LLMs as generalist embedding models](#). In *The Thirteenth International Conference on Learning Representations*.
- Jinhyuk Lee, Feiyang Chen, Sahil Dua, Daniel Cer, Madhuri Shanbhogue, Iftekhhar Naim, Gustavo Hernández Ábrego, Zhe Li, Kaifeng Chen, and Henrique Schechter Vera. 2025b. Gemini embedding: Generalizable embeddings from gemini. *arXiv preprint arXiv:2503.07891*.
- Xiaopeng Li, Xiangyang Li, Hao Zhang, Zhaocheng Du, Pengyue Jia, Yichao Wang, Xiangyu Zhao, Huifeng Guo, and Ruiming Tang. 2024. Syneg: Llm-driven synthetic hard-negatives for dense retrieval. *arXiv preprint arXiv:2412.17250*.
- Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *Cornell University - arXiv, Cornell University - arXiv*.
- Zihe Liu, Jiashun Liu, Yancheng He, Weixun Wang, Jiaheng Liu, Ling Pan, Xinyu Hu, Shaopan Xiong, Ju Huang, and Jian Hu. 2025. Part i: Tricks or traps? a deep dive into rl for llm reasoning. *arXiv preprint arXiv:2508.08221*.
- 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.
- Yu A Malkov and Dmitry A Yashunin. 2018. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence*, 42(4):824–836.
- Bhaskar Mitra and Nick Craswell. 2017. Neural models for information retrieval. *arXiv: Information Retrieval, arXiv: Information Retrieval*.
- Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. [MTEB: Massive text embedding benchmark](#). In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037. Association for Computational Linguistics.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, and Alex Ray. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, and Jack Clark. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmlR.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-BERT: Sentence embeddings using siamese BERT-networks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992. Association for Computational Linguistics.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv: Learning, arXiv: Learning*.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, and Yang Wu. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.
- Shuhe Wang, Shengyu Zhang, Jie Zhang, Runyi Hu, Xiaoya Li, Tianwei Zhang, Jiwei Li, Fei Wu, Guoyin Wang, and Eduard Hovy. 2024. Reinforcement learning enhanced llms: A survey. *arXiv preprint arXiv:2412.10400*.
- Weixun Wang, Shaopan Xiong, Gengru Chen, Wei Gao, Sheng Guo, Yancheng He, Ju Huang, Jiaheng Liu, Zhendong Li, and Xiaoyang Li. 2025. Reinforcement learning optimization for large-scale learning: An efficient and user-friendly scaling library. *arXiv preprint arXiv:2506.06122*.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, and Chenxu Lv. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, and Junyang Lin. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

A Related Work

A.1 Dense Retrieval

Dense retrieval encodes queries and items into a shared semantic space and retrieves the top-N most similar items using approximate nearest neighbor (ANN) algorithms (Malkov and Yashunin, 2018). Early dense retrieval methods primarily relied on encoder-only pre-trained models like BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019). Gao et al. (2021) first introduced a contrastive learning framework for dense retrieval tasks, using dropout as an unsupervised self-prediction technique in its contrastive learning framework. After that, the contrastive learning paradigm became widely adopted. Beyond pure text representation, CLIP (Radford et al., 2021) leverages a large-scale image-text matching dataset to pretrain multimodal representations through simple contrastive learning objectives for multimodal alignment tasks. With the rapid development of large language models, many studies now employ LLMs as backbones for dense retrieval, leveraging their extensive world knowledge and zero-shot generalization capabilities to obtain superior universal semantic representations. Notably, RepLLaMA (Ma et al., 2024) first demonstrated the effectiveness of multi-stage supervised fine-tuning (SFT) of large language models for text retrieval. NV-Embed (Lee et al., 2025a) proposes a latent attention layer to obtain pooled embeddings and enhances model performance through contrastive learning with in-batch negatives and curated hard negative examples. Gemini Embedding (Lee et al., 2025b) utilizes Gemini’s multilingual and code comprehension capabilities to generate highly versatile embeddings across languages and text modalities. Seed Embedding (ByteDance Seed, 2025) adopts a Mixture-of-Experts (MoE) model as its backbone, supporting multiple embedding dimensions via Matryoshka Representation Learning (MRL) (Kusupati et al., 2022) training similar to NV-Embed, and employs two-stage contrastive learning to strengthen universal representation capabilities. The gte-Qwen (Li et al., 2023) integrates bidirectional attention mechanisms and enhances performance through instruction tuning using both weakly supervised and supervised data. Building on gte-Qwen, Qwen3 Embedding (Zhang et al., 2025) introduces an innovative multi-stage training pipeline that combines large-scale unsupervised pre-training with supervised fine-tuning on high-quality datasets, ensuring robustness and

adaptability through effective model merging strategies. Current LLM-based dense retrieval models achieve state-of-the-art (SOTA) performance on the MTEB leaderboard (Muennighoff et al., 2023), demonstrating the effectiveness of large models for semantic representation.

A.2 Reinforcement Learning for LLM

In recent years, large language model (LLM) technology has rapidly advanced, with reinforcement learning (RL) (Wang et al., 2024) being widely integrated into LLM training pipelines as a key technical approach to enhance model performance and controllability. The core application of RL in LLMs involves modeling text generation as a sequential decision-making problem, where the model acts as an agent, generated text represents actions, and external feedback forms reward signals. Among these approaches, Reinforcement Learning from Human Feedback (RLHF) has quickly become the dominant paradigm. OpenAI’s Proximal Policy Optimization (PPO) (Schulman et al., 2017) stands as one of the most representative algorithms in the RLHF framework. Subsequently, OpenAI first applied PPO to align human preferences in its InstructGPT (Ouyang et al., 2022). In PPO, the policy model serves as the optimization target, generating sampled text from input prompts. The reward model, trained on human preference data, provides scalar reward signals based on response quality from the policy model. A reference model calculates KL divergence penalties to prevent excessive deviation from the original model, while a value model estimates action value functions representing expected cumulative future value. The PPO algorithm coordinates these four components to align human preferences. Despite PPO’s success, its complex workflow and high computational costs persist. The Direct Preference Optimization (DPO) (Rafailov et al., 2023) algorithm addresses this by assuming human preferences follow the Bradley-Terry model, simplifying preference learning into a directly optimizable classification problem through maximum likelihood objectives. While DPO achieves stable performance with higher efficiency and outperforms PPO in some tasks, PPO maintains advantages in complex multi-step reasoning tasks. To reduce PPO’s complexity and computational demands, DeepSeek proposes the GRPO (Shao et al., 2024) algorithm during the training of its DeepSeekMath. GRPO eliminates the value model in PPO and adopts a group-



Figure 4: Recall comparison between our model and the online baseline on four long-tail query-slices

sampling-based baseline estimation method, using average group rewards as the baseline to calculate inter-group advantages, significantly reducing computational overhead. Building on PPO, Liu et al. (2025) discovered that simply applying robust advantage normalization and token-level loss aggregation techniques to vanilla PPO could consistently improve performance. Currently, the RL research continues to drive progress in LLM training (Guo et al., 2025), providing sustainable iterative methods for aligning large language models with human preferences.

Most prior RL methods above optimize sequential text generation, whereas hard-negative refresh methods for retrieval still construct binary examples in a separate mining stage. Retrieval-GRPO instead casts dense retrieval as a single-step decision: the current retriever’s own top- k outputs are refreshed at every update, without an offline mining stage or index rebuild, and a frozen relevance judge assigns a continuous reward to every candidate. Our optimizer is a group-normalized policy gradient with KL regularization; the contribution is this retrieval reformulation and its production-scale validation, rather than a new general-purpose optimizer.

B Implementation Details

The model is trained on a cluster of 128 GPUs (each equipped with 96GB memory). The training framework is implemented based on ROLL (Wang et al., 2025), an open-source distributed RL-training library optimized for large language models. During the Supervised Fine-Tuning (SFT) phase, a

per-GPU batch size of 256 is configured with the AdamW optimizer ($\beta_1=0.9$, $\beta_2=0.95$), a learning rate of $1e-5$, and a cosine learning rate scheduler incorporating a warmup ratio of 0.05. The architecture enforces a maximum input length of 256 tokens, with query and item embeddings projected into a 1024-dimensional L2-normalized space. The temperature parameter τ is set to 0.02 for the InfoNCE contrastive loss. MRL is applied across dimensions [1024, 2048, 3072] to enhance hierarchical representation learning. This phase completes after 1 full training epoch over the dataset.

Transitioning to the Retrieval-GRPO phase, the policy model (3B parameters) interacts with the TaoSR1 as reward model—a 42B Mixture-of-Experts (MoE) architecture activating 3.5B parameters per instance. Maintaining consistency with the SFT phase, the training protocol employs a per-GPU batch size of 256, AdamW optimization ($\beta_1=0.9$, $\beta_2=0.95$), and a reduced learning rate of $1e-6$, coupled with the same cosine scheduler and 256-token context window. For candidate selection, items are gathered across 128 devices to form a candidate pool of $|\hat{B}| = 128 \times 256 = 32,768$ items, from which the top- $k = 8$ candidates are selected per query for reward computation. The temperature parameter τ in the policy distribution (Equation 3) is set to 1, producing a softer distribution over candidates that allows all top- k items to contribute meaningful gradients. The KL divergence penalty coefficient ($\beta=0.05$, as defined in Equation 5) regulates policy deviation from reference behavior. Table 5 shows that removing KL degrades both test sets, while the $\beta \rightarrow \infty$ endpoint

reduces analytically to the SFT reference. We did not tune $\beta = 0.05$ on this task and therefore do not claim it is optimal. Reducing the candidate pool from 32,768 (128 GPUs) to 16,384 (64 GPUs) changes both Hitrate@6k metrics by less than 0.5 points, showing that the result is not tied to the largest pool.

β	General	Long-Tail
0 (no KL)	48.37	40.85
0.05 (ours)	49.35	41.07
$\rightarrow \infty$ (SFT)	48.00	37.52

Table 5: KL-coefficient ablation on Hitrate@6k.

Embedding dimensions remain fixed at 1024 to ensure compatibility with upstream components. The GRPO phase iterates over 3 epochs to achieve convergence.

C Effect of Reward Model

To validate the role of reward model quality in Retrieval-GRPO, we replace the default TaoSR1 (a 42B-MoE relevance model activating 3.5B parameters) with two smaller alternatives: a 4-layer BERT-based relevance model and a 13B dense relevance model. As shown in Table 6, R-GRPO’s effectiveness scales with reward model quality. The 4-layer BERT model provides noisy relevance signals that mislead policy optimization, resulting in degraded performance compared to the SFT baseline. The 13B model provides sufficiently accurate rewards to achieve substantial improvements, particularly on long-tail queries. The 42B MoE model further pushes performance. This demonstrates that precise relevance calibration is pivotal for R-GRPO: inaccurate reward signals not only fail to help but can actively harm the learned representations.

Model	Hitrate@6k		Goodrate@100
	General	Long-Tail	
SFT	48.00	37.52	63.44%
SFT + R-GRPO(ReI-BERT-4l)	47.40	37.22	63.67%
SFT + R-GRPO(ReI-13B)	48.88	40.56	65.03%
SFT + R-GRPO(ReI-42BA3.5B)	49.35	41.07	65.91%

Table 6: Ablation Results on Reward Model

D Additional Case Studies

Figure 4 compares the recall results of our model with the online baseline across four long-tail query slices. Figure 5 presents additional examples of top-k items and relevance scores during R-GRPO

training, and Figure 6 provides further recall comparisons on long-tail queries.



Figure 5: Additional examples of top-k items and relevance scores assigned by the reward model during R-GRPO training

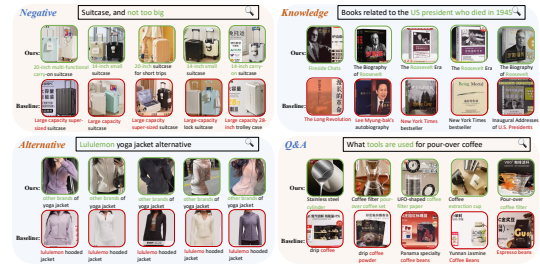


Figure 6: Additional recall comparison between our model and the online baseline on long-tail queries