

RAQE: Reranker-Aligned Query Expansion via Label-Free Group-Relative Policy Optimization

Gyeonghun Sun¹, Jeonghwan Choi¹, Sundong Kim², Hwanjun Song^{1,3,*}

¹Korea Advanced Institute of Science and Technology (KAIST)

²Gwangju Institute of Science and Technology (GIST), ³Cluvion

{skhoon4u, songhwanjun}@kaist.ac.kr

Abstract

Retrieval-augmented generation is often bottlenecked by retrieval quality, motivating query expansion. We propose RAQE (Reranker-Aligned Query Expansion), a label-free framework that trains a compact generator (3B/8B) for query expansion via policy optimization using a reranker-derived reward. RAQE replaces human relevance labels with Reranker-Shaped Discounted Gain (RSDG), computed from reranker scores over the retriever’s top- k documents. For query expansion, RAQE generates a single, short expansion in one pass, avoiding long outputs and repeated generation while remaining low latency. Across eight IR benchmarks, RAQE is competitive with strong large language model prompting baselines and narrows the gap to reinforcement learning methods that rely on costly human supervision. The code is available at <https://github.com/DISL-Lab/RAQE>.

1 Introduction

Retrieval-augmented generation (RAG) improves factuality and coverage of large language models (LLMs) by grounding generation in retrieved evidence (Lewis et al., 2020). However, retrieval quality is often limited by lexical mismatch and query underspecification: real user queries are short, ambiguous, and often lack direct lexical overlap with relevant passages. To address this limitation, query expansion (QE) emerges as a principled approach that augments the original query with additional terms or contextual information, building on a long-standing tradition in information retrieval (IR) (Lavrenko & Croft, 2001; Carpineto & Romano, 2012). Specifically, existing QE methods can be categorized into two representative directions. First, recent approaches such as MuGI (Zhang et al., 2024) and Word2Passage (Choi et al., 2025) adopt a *training-free* paradigm, generating pseudo-passages directly from off-the-shelf LLMs without task-specific post-training. In contrast, another line of work, such as ExpandR (Yao et al., 2025) and OPQE (Xu et al., 2025), adopts reinforcement learning (RL) to avoid repeated generation by training a model to produce a *single, high-quality* query expansion aligned with retrieval objectives.

Despite their different design choices, the two paradigms induce a fundamental trade-off that limits their practical utility in QE. Training-free methods avoid human supervision¹ but suffer from high inference latency due to repeated pseudo-passages generation, while RL-based methods reduce inference cost via a single expansion but require supervised relevance signals to define retrieval-aligned rewards, which are often costly or infeasible in cold-start domains. As a result, this trade-off remains a key bottleneck for deploying QE in real-world RAG systems, as it forces practitioners to choose between high latency at inference and costly supervision for post-training, despite their significant IR performance gains over other post-retrieval techniques, such as re-ranking (Wang et al., 2025; Abdallah et al., 2025) or compression (Song et al., 2026; Xu et al., 2023).

*Corresponding author.

¹Human supervision refers to query–document relevance labels to assess whether an expanded query improves retrieval, which are costly to obtain at scale given the size of modern document corpora over 1M (Rassin et al., 2024; Takehi et al., 2025).

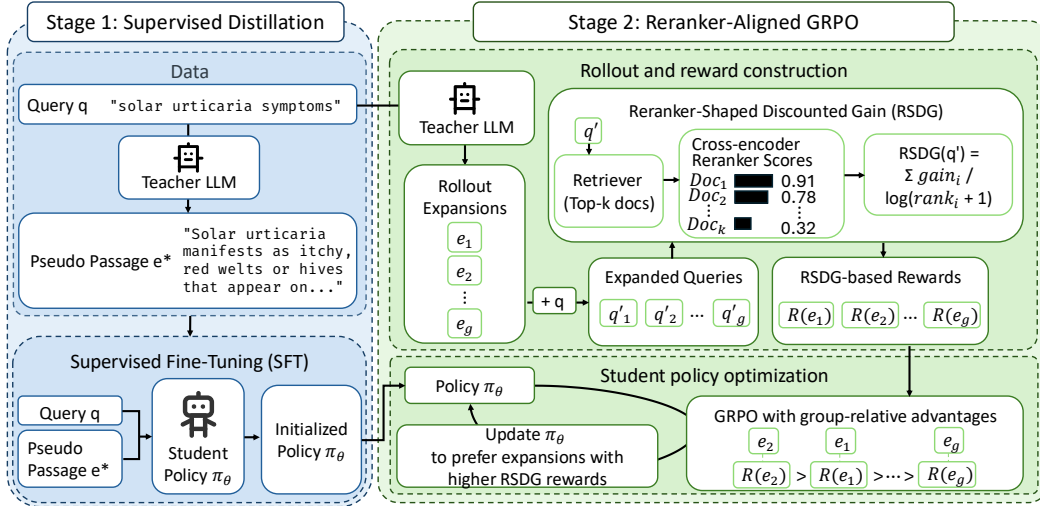


Figure 1: Overview of RAQE: Two-phase label-free training framework for QE, adopting RSDG-driven preference signals for optimization without relying on human labels.

In this paper, we move beyond optimizing retrieval accuracy alone and instead aim to realize a truly practical QE framework, explicitly accounting for two critical system-level constraints: “supervision cost” during training and “inference latency” at deployment. To achieve this objective, we propose RAQE (Reranker-Aligned Query Expansion) in Figure 1, a label-free group-relative policy optimization framework for QE that eliminates human supervision, while also achieving fast inference by generating an effective pseudo-passage in a single forward pass. Our key idea is to use reranker scores as a *proxy* for the query-chunk relevance score, transforming them into relative preference signals that guide group-relative policy optimization for QE. This enables high-quality query expansion with a compact QE model while eliminating supervision cost and reducing inference latency.

Specifically, we formulate query expansion as a *relative decision* problem, where the objective is not to estimate absolute query-chunk relevance, but to learn which expansions yield greater retrieval gains than others for the same query. To this end, we introduce a *distill-then-optimize* framework: an initial distillation phase using Supervised Fine-Tuning (SFT), shown on the left side of Figure 1, that encourages the model to generate concise and well-formed query expansions, followed by label-free group-relative policy optimization guided by reranker-induced relative preference signals, shown on the right side of Figure 1. Particularly in the second phase, reranker scores define a preference structure over rollouts (*i.e.*, pseudo-passages forming candidate expansions) based on their downstream retrieval impact. We instantiate this signal via *Reranker-Shaped Discounted Gain (RSDG)*, a relative measure of expansion quality derived from reranker scores over retrieved documents. Notably, RSDG exhibits strong alignment with human supervision (see Table 3), allowing it to serve as a reliable supervision signal. As a result, RSDG-induced preferences guide the policy toward effective expansions without any human annotation.

We conduct extensive experiments to evaluate the effectiveness and practicality of RAQE. Across eight IR benchmarks, covering five in-domain and three out-of-domain settings, RAQE consistently delivers superior retrieval quality over strong LLM-based query expansion baselines, while preserving low inference latency via single-pass generation. We further show that the proposed RSDG reward provides reliable preference signals that closely align with human label-based retrieval metrics, enabling stable policy optimization without human supervision. Finally, ablation studies confirm that the proposed distill-then-optimize framework plays a complementary role, where distillation provides a strong initialization and reranker-aligned optimization further improves retrieval effectiveness.

Our main contributions are summarized as follows: (1) We resolve a fundamental trade-off in prior query expansion methods between supervision and latency, enabling label-free training with single-pass inference; (2) We propose a distill-then-optimize framework, RAQE,

for training compact query expansion models via group-relative policy optimization; and (3) We introduce RSDG, a reranker-derived proxy reward that provides human-aligned training signals, enabling effective optimization without manual relevance labels.

2 Related Work

Query expansion (QE) has been revisited from two complementary directions using LLMs: training-free generation and RL-based alignment of expansion policies. Beyond classical QE methods (Carpineto & Romano, 2012; Rocchio, 1971; Lavrenko & Croft, 2001), recent LLM-based approaches shift QE toward generating richer contextual text that can better match retriever representations, mitigating vocabulary mismatch in sparse retrieval and bridging the structural gap in dense embedding spaces (Gao et al., 2023; Wang et al., 2023; Zhang et al., 2024; Choi et al., 2025; Rafailov et al., 2023).

Classical Query Expansion. Early QE methods were primarily based on pseudo-relevance feedback (PRF), where top-ranked documents from an initial retrieval pass are assumed to be relevant and used to extract expansion terms (Carpineto & Romano, 2012). Representative approaches include Rocchio-style vector updates (Rocchio, 1971) and probabilistic models such as RM3 (Lavrenko & Croft, 2001). However, these classical methods typically augment queries with isolated keywords rather than coherent text, limiting their ability to capture richer semantic relationships or bridge vocabulary mismatches in dense retrieval settings.

Training-Free Methods. With the emergence of LLMs, recent work revisits QE in a training-free manner using LLMs by generating pseudo-passages to enrich query representations. Unlike PRF, they do not depend on top-ranked documents from the target retriever, but instead rely solely on the generative prior of large language models. Specifically, HyDE (Gao et al., 2023) generates hypothetical documents to improve dense retrieval. Query2Doc (Wang et al., 2023) produces a single pseudo-passage concatenated with the original query. MuGI (Zhang et al., 2024) aggregates multiple pseudo-passages for robustness, and Word2Passage (Choi et al., 2025) generates structured outputs (word, sentence, and passage) for fine-grained weighting. These variants highlight a trade-off between robustness and inference cost, as multi-sample generation and structured outputs increase latency. Accordingly, many training-free methods improve retrieval by increasing the generation budget (e.g., more samples or larger generators) rather than explicitly optimizing the generation policy for retrieval objectives, leaving the generator weakly aligned with ranking signals.

RL-Based Methods. Another line of work aligns query expansion with ranking-oriented objectives via RL, including PPO (Schulman et al., 2017) and DPO (Rafailov et al., 2023). In detail, ExpandR (Yao et al., 2025) optimizes expansions using rank-based and answer-based rewards, while OPQE (Xu et al., 2025) studies on-policy optimization for query expansion. Optimizing directly for ranking metrics such as NDCG provides a more principled objective than heuristic prompt engineering. Despite reduced latency from single-passage generation, these methods rely on human-annotated query–document relevance labels. This reliance typically appears through supervised reward modeling or oracle relevance judgments, which may limit scalability across domains where labels are scarce or expensive. Furthermore, training reward models with human labels introduces additional annotation cost and may entangle QE performance with dataset-specific supervision.

Label-Free Reward Design for Retrieval. Our work is also related to recent efforts that leverage proxy signals to approximate ranking supervision without explicit human labels. Cross-encoder rerankers, trained on relevance data, have been widely used to refine retrieval outputs. Several works have explored distilling reranker knowledge into bi-encoders or retrievers (Qu et al., 2021; Lin et al., 2021; Hofstätter et al., 2021). More recently, reranker-derived signals have also been explored for optimizing query reformulation without explicit relevance annotations (Mao et al., 2024). In this work, reranker scores are used not to supervise the retriever directly, but to shape the generation policy of a query expansion model via group-relative preference signals. This design approximates ranking supervision via reranker-derived discounted gains without human relevance labels.

3 Reranker-Aligned Query Expansion

Problem formulation. Given a user query q , our goal is to generate a single, concise pseudo-passage e and construct an expanded query $q' = \text{concat}(q, e)$ for retrieval. A fixed retriever (e.g., BM25 or a dense retriever) is then applied to retrieve the top- k documents $D_{q'} = \{d'_1, \dots, d'_k\}$, which are subsequently used to augment the input prompt of the LLM serving as the response generator in the RAG pipeline.

We train a conditional policy $\pi_\theta(e | q)$ to generate expansions that improve the retrieval ranking induced by q' . Unlike iterative or multi-generation QE, the policy produces one expansion in a single forward pass. The central challenge is obtaining retrieval-aligned supervision without costly human relevance annotations.

3.1 Reranker-Shaped Discounted Gain (RSDG)

RAQE aligns query expansions with retrieval quality without human relevance labels, which are normally required to compute ranking metrics such as NDCG. Instead, it replaces label-based supervision with a reranker-derived signal, using cross-encoder scores as a proxy for relevance. This proxy is used twice in the framework: first, to curate distillation data by retaining only queries where the teacher expansion improves retrieval, and second, to define the scalar reward for GRPO that guides optimization (details are in Section 3.2). This design allows us to approximate ranking-oriented supervision while avoiding the need for costly human-labeled relevance judgments. Importantly, the reranker is used only to compute training signals; it is not required at inference time.

Modification of NDCG to RSDG. To construct a label-free reward signal, we modify the standard NDCG (Wang et al., 2013) by replacing human relevance labels with reranker-derived gains. Concretely, given an expanded query q' , we score its top- k retrieved documents, $D_{q'} = \{d'_1, \dots, d'_k\}$, using a cross-encoder (CE) reranker² conditioned on the original query, and treat these scores as a proxy for relevance:

$$s(q, d'_i) = \max(\text{CE}(q, d'_i), 0), \text{ where} \quad (1)$$

negative scores are clamped to zero, as they indicate the absence of relevance evidence. This transformation ensures non-negative gains, preserving the monotonicity assumption of DCG while discarding negative relevance signals that could destabilize optimization. Using the non-negative CE score, we compute a DCG-style aggregation over the retriever’s ranking, and normalize it by the ideal ranking induced by the reranker scores (serving as a surrogate for IDCG). This yields a Reranker-Shaped Discounted Gain (RSDG):

$$\text{RSDG}@k(q', D_{q'}) = \frac{\sum_{i=1}^k s(q, d'_i) / \log_2(i+1)}{\sum_{j=1}^k s(q, d'_{(j)}) / \log_2(j+1)}, \text{ where} \quad (2)$$

i indexes the retriever ranking for q' , and j indexes the same documents sorted by $s(q, d')$. Notably, reranker scores are computed with respect to the original query q rather than the expanded query q' . This prevents the reward from trivially favoring expansions that restate generated content, and instead anchors relevance estimation to the user’s true information need. The denominator corresponds to an idealized ranking induced by the reranker scores, serving as a surrogate for IDCG. This normalization constrains RSDG@ k to the $[0, 1]$ range and makes it comparable across queries. Thus, higher RSDG values reflect stronger retrieval alignment of the expanded query, achieved without relying on human relevance labels.

Intuitively, RSDG measures how well the ranking induced by the expanded query q' aligns with the reranker’s preference ordering. If the expanded query leads the retriever to rank documents in a way that agrees with the reranker scores, the numerator and denominator become similar, yielding a high RSDG value and therefore a higher reward for the expansion. Conversely, if documents that receive high reranker scores appear lower in the ranking produced by q' , the discounted gain decreases, making the expansion less preferred.

²We use BGE-Reranker (BAAI/bge-reranker-v2-m3) as the default reranking model, and show in Section 4.2 that our performance gains for QE persist across other rerankers as well.

3.2 Distill-then-Optimize Framework

We train our policy model $\pi_\theta(e \mid q)$ in two phases: (i) *distillation*, which learns the format and content of effective expansions; and (ii) *reranker-aligned GRPO*, which aligns expansions with retrieval effectiveness using the RSDG score as a proxy reward. The two phases play complementary roles: Phase 1 provides linguistic and structural priors for well-formed pseudo-passages, while Phase 2 explicitly aligns generation with ranking improvements. Directly optimizing RSDG from scratch is challenging because the reward depends on discrete retrieval rankings and is therefore highly non-smooth. The distillation phase reduces this difficulty by constraining the initial policy to a manifold of linguistically coherent expansions, thereby narrowing the search space explored during RL fine-tuning.

3.2.1 Phase 1: Supervised Distillation

This phase provides a strong initialization for the QE policy. Using supervised distillation from a strong teacher model, the model learns the structure and content of effective query expansions, providing a stable starting point. Here, we focus on the construction of distillation data for training. The optimization itself follows a conventional SFT objective, which can be viewed as a sequence-level distillation of the teacher model.

Data Preparation. We construct distillation data using the official training splits of five existing benchmark datasets: NFCorpus (Boteva et al., 2016), FiQA (Maia et al., 2018), FEVER (Thorne et al., 2018), HotpotQA (Yang et al., 2018), and MS MARCO (Bajaj et al., 2016). Together, they span diverse domains and query types, exposing the generator to heterogeneous query intents during training and thereby improving robustness while reducing overfitting to dataset-specific lexical patterns.

For each dataset, we sample 3,000 queries uniformly at random to construct an initial pool of training data. Next, for each query q in the pool, we collect a potentially desirable pseudo-passage e^* using a strong LLM³ with the prompt in Appendix G. Each pair (q, e^*) serves as a supervision signal for supervised fine-tuning (SFT).

Data Curation. However, teacher expansions do not necessarily translate into improvements in actual retrieval performance, and naive usage may amplify noisy generations. We therefore filter training queries based on retrieval impact, as including expansions that degrade retrieval would introduce contradictory supervision. Specifically, we measure the retrieval improvement $\Delta(q', q)$ defined as:

$$\Delta(q', q) = \text{RSDG}@k(q', D_{q'}) - \text{RSDG}@k(q, D_q), \text{ where } q' = \text{concat}(q, e). \quad (3)$$

Intuitively, this measures how much the expanded query improves the ranking quality compared to the original query. Only queries with $\Delta(q', q) > 0$ are retained. Empirically, teacher-generated expansions occasionally introduce off-topic entities or spurious associations. By retaining only queries with positive $\Delta(q', q)$, we ensure that the distilled supervision is consistent with ranking improvements, preventing the policy from imitating expansions that harm retrieval performance. Dataset statistics are provided in Appendix B.

Optimization. We warm-start the generator π_θ by minimizing the negative log-likelihood of the teacher pseudo-passage. Formally, the SFT objective is:

$$\mathcal{L}_{\text{SFT}}(\theta) = -\mathbb{E}_{(q, e^*)} \log \pi_\theta(e^* \mid q). \quad (4)$$

3.2.2 Phase 2: Reranker-Aligned GRPO

We further optimize π_θ using reranker-aligned GRPO from the distilled checkpoint obtained in Phase 1. We reuse the same filtered training queries from Phase 1.

Rollout Construction and Reward Modeling. For each training query q , we construct a fixed offline candidate set (rollouts) $\{e_1, \dots, e_g\}$ using the teacher. To avoid the non-stationarity and computational overhead of on-the-fly retrieval during training, we pre-compute and cache the retrieval results and reranker scores for these candidates. This

³We use Llama-3.1-70B-Instruct as the teacher to obtain a strong starting point from Phase 1, but the choice is not essential and can be replaced by other strong LLMs.

pre-computation effectively acts as an offline candidate reweighting mechanism, stabilizing advantage estimation compared to on-policy learning, particularly when training compact policy models (Lai et al., 2025; Lanchantin et al., 2025).

As in Phase 1, the reward of each expansion e in the candidate set is computed as $r_i = \Delta(q', q)$ based on RSDG@ k in Eq. (2), which captures the RSDG@ k improvement of the expanded query over the original. Then, rewards are standardized within each rollout group to compute group-relative advantages, A_i , which reduces variance and focuses learning on relative preference ordering rather than absolute score magnitude. Using group-relative advantages transforms the problem from absolute reward maximization to preference-based optimization. Although $\Delta(q', q)$ is a proxy reward and does not rely on explicit human relevance labels, it can still provide reliable ordering signals among candidate expansions, as evidenced by the high agreement with human preference (see Table 3). Consequently, the policy only needs to rank candidate expansions correctly within each rollout group rather than rely on globally calibrated reward magnitudes.

Optimization. We optimize the policy π_θ using the standard GRPO objective (Shao et al., 2024). Concretely, let $\hat{A}_i$ denote the standardized advantage of candidate e_i within its rollout group. The clipped surrogate objective is:

$$\mathcal{L}_{\text{GRPO}} = -\frac{1}{g} \sum_{i=1}^g \min\left(\rho_i \hat{A}_i, \text{clip}(\rho_i, 1 - \epsilon, 1 + \epsilon) \hat{A}_i\right), \text{ where } \hat{A}_i = \frac{r_i - \bar{r}}{\sigma_r} \quad (5)$$

and $\rho_i = \exp(\log \pi_\theta(e_i|q) - \log \pi_{\text{ref}}(e_i|q))$ is the likelihood ratio with respect to a frozen reference policy, and $r_i = \Delta(q'_i, q)$ is the RSDG-based reward defined in Eq. (3), with $\bar{r}$ and σ_r denoting the mean and standard deviation of rewards within the rollout group. This objective encourages the policy to increase the probability of expansions that improve RSDG while maintaining proximity to the distilled initialization.

4 Evaluation

We compare RAQE to six LLM-based QE baselines: *Default*, which retrieves documents using the original query without expansion; four training-free QE methods (*HyDE*, *Query2Doc*, *MuGI*, and *Word2Passage*); and an RL-based QE method, *ExpandR*, regarded as a strong label-supervised reference among RL-based approaches due to its reliance on oracle (human) relevance labels. All QE methods are evaluated on the test split of eight datasets spanning diverse domains, including MS MARCO (Bajaj et al., 2016) and seven BEIR benchmarks (Thakur et al., 2021). The test set includes five *in-distribution* datasets (NFCorpus, FiQA, FEVER, HotpotQA, MS MARCO) and three *out-of-distribution* datasets (ArguAna, NQ, SciDocs) to evaluate RAQE’s generalization. All the prompt templates are in Appendix G.

Training. For RAQE, we mainly use Llama-family backbones (3B and 8B) with LoRA-based fine-tuning and set a rollout group size of $g = 8$ by default (see Appendix E.1 for ablation). Training is highly efficient, completing in under one hour on two NVIDIA RTX Pro 6000 GPUs. Further implementation details are provided in Appendix A.

Evaluation. We evaluate RAQE on two aspects: retrieval quality and downstream task performance, respectively. For retrieval evaluation in Sections 4.1, we retrieve $k=10$ documents using a sparse retriever (BM25) and a dense retriever (Contriever (Izacard et al., 2022)), and report NDCG@10 with human relevance labels. For downstream evaluation in Section 4.3, we assess task performance on open-domain question answering and fact verification to measure how retrieval improvements translate to end tasks.

4.1 Improved Retrieval and Reduced Latency

In this section, we evaluate RAQE via two experiments: (1) retrieval quality across sparse and dense settings, and (2) the quality–latency trade-off across backbone sizes.

Quality Comparison. Tables 1 and 2 report NDCG@10 for sparse (BM25) and dense (Contriever) retrieval. In general, RAQE achieves the highest NDCG@10 among methods with the same backbone size (8B) and outperforms counterparts using substantially larger models

Method	Gen.	NFC	FiQA	FEV	HP	MSM	Arg	NQ	SciD
ExpandR (Label-Supervised Reference)	8B	35.6	26.1	84.2	69.7	66.4	28.6	43.1	15.1
Default (Original)	–	32.2	23.5	67.9	63.9	50.6	27.7	27.0	14.0
HyDE	8B	34.2	18.9	82.2	60.5	58.1	27.2	42.6	14.0
(Gao et al., 2023)	70B	34.8	20.7	82.3	67.3	61.8	28.1	45.8	13.4
Query2Doc	8B	32.8	23.5	79.6	65.7	57.8	26.8	37.5	14.2
(Wang et al., 2023)	70B	34.1	24.1	79.1	66.7	59.9	26.9	40.4	14.4
MuGI	8B	28.5	20.2	81.4	48.7	51.4	27.6	31.3	13.1
(Zhang et al., 2024)	70B	34.4	20.8	82.7	67.5	60.4	27.3	45.6	13.2
Word2Passage	8B	35.0	20.4	79.4	70.1	61.4	26.5	41.3	9.4
(Choi et al., 2025)	70B	22.7	20.0	83.5	67.8	58.7	24.1	43.9	9.2
RAQE (ours)	8B	36.0	24.5	84.5	69.1	65.5	28.3	44.4	14.7

Table 1: NDCG@10 for sparse retrieval (BM25) across eight datasets (0–100 scale). Baselines use Llama-3.1-8B/70B-Instruct, but RAQE uses Llama-3.1-8B-Instruct. The left five datasets are in-domain (used for training), while the right three are out-of-domain.

Method	Gen.	NFC	FiQA	FEV	HP	MSM	Arg	NQ	SciD
ExpandR (Label-Supervised Reference)	8B	33.5	29.2	81.6	63.0	57.1	31.8	38.1	16.5
Default (Original)	–	31.8	24.2	68.6	45.4	42.6	26.2	22.5	14.5
HyDE	8B	29.7	19.5	79.3	50.1	61.1	21.9	32.8	15.5
(Gao et al., 2023)	70B	33.0	27.4	79.1	59.9	59.4	19.8	39.2	14.9
Query2Doc	8B	26.0	19.0	60.5	34.6	44.3	15.3	25.5	14.5
(Wang et al., 2023)	70B	27.4	21.6	62.6	39.7	42.8	13.8	26.6	13.2
MuGI	8B	22.0	15.2	47.8	22.4	51.3	13.5	19.8	9.3
(Zhang et al., 2024)	70B	22.7	14.7	46.5	29.3	54.4	13.5	24.0	9.6
Word2Passage	8B	20.0	14.4	56.2	40.7	24.9	12.1	18.6	6.2
(Choi et al., 2025)	70B	16.1	13.7	55.2	44.6	45.7	12.7	24.5	5.8
RAQE (ours)	8B	33.5	28.8	80.9	62.9	56.9	28.5	40.9	16.4

Table 2: NDCG@10 when using dense retrieval (Contriever) across eight datasets (0–100 scale). Same evaluation protocol as Table 1.

(70B). While most training-free QE methods are primarily designed for sparse retrieval and often yield limited gains or even degradation under dense retrieval due to lexical expansion mismatch, RAQE consistently improves performance in both settings through retrieval-aligned optimization. These gains are consistent across both in-domain (3rd–7th columns) and out-of-domain (8th–10th columns) settings, indicating strong cross-domain generalization. Moreover, RAQE achieves performance on par with, or slightly exceeding, the strong label-supervised ExpandR reference. Together, these results show that RSDG effectively captures retrieval-relevant preferences without human supervision.

Quality vs Latency Trade-Off. Figure 2 shows the retrieval quality–latency trade-off as the Llama3 backbone is scaled from 3B to 70B. RAQE achieves the strongest retrieval improvements with only a modest increase in latency. In contrast, HyDE, MuGI, and Word2Passage incur substantial latency overheads due to repeated generation, while Query2Doc avoids repeated generation but primarily relies on scaling the generator up to 70B, yielding limited retrieval gains. This indicates that model scaling alone is insufficient, while reranker-aligned optimization more effectively converts model capacity into retrieval gains. The results with Qwen3-4B/30B are in Appendix F.2.

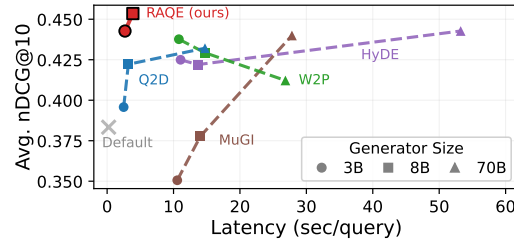


Figure 2: Retrieval quality–latency trade-off when scaling the Llama3 backbone from 3B to 70B using BM25. Both latency and NDCG@10 are macro-averaged across 8 datasets. The detailed results of RAQE with 3B are in Section F.1.

Variant	NFC	FiQA	FEV	HP	MSM	Arg	NQ	SciD
Default (no training)	32.2	23.5	67.9	63.9	50.6	27.7	27.0	14.0
Stage 1 only (distillation)	35.1	25.5	78.8	68.3	61.7	28.0	41.1	14.1
Stage 2 only (GRPO)	35.3	25.5	82.5	65.7	60.3	27.9	38.4	14.0
RAQE (ours)	36.0	24.5	84.5	69.1	65.5	28.3	44.4	14.7

Table 5: Stage-wise ablation of RAQE under BM25: “Stage 1 only” corresponds to the setup using supervised distillation only, while “Stage 2 only” applies GRPO-based optimization without the distillation stage.

4.2 Comprehensive Analysis of RAQE

In this section, we evaluate (1) the validity of the RSDG reward by comparing it with human judgments, (2) the impact of improved retrieval performance on generation quality, and (3) the robustness of RAQE to different reranker models.

Validation of RSDG Reward. We validate whether the RSDG reward in Eq. (2) provides a correct preference signal. For each training query, we form rollout groups ($g=8$) and evaluate alignment in two ways. First, we randomly sample pairs of candidate expansions within each group and measure whether RSDG@10 and human-labeled NDCG@10 induce the same preference ordering, reporting the pairwise agreement rate. Second, we assess the global consistency of preferences over the full rollout group using correlation metrics (Spearman and Pearson). For the former, Table 3 shows that RSDG achieves high pairwise agreement, substantially exceeding the random baseline (50%). For the latter, Spearman and Pearson correlations confirm high agreement in relative ordering and reward trends, supporting RSDG as an effective label-free signal. Together, these results support RSDG as an effective label-free signal for policy optimization.

	Pairwise	Spearman	Pearson
Human Alignment	0.730	0.431	0.455

Table 3: Alignment between label-free RSDG@10 and label-based NDCG@10. Results are macro-averaged over five in-domain datasets under BM25 retrieval.

Comparison with a RaFe-style Reward. To isolate the contribution of the proposed reward formulation, we replace RSDG with a flat averaged reranker-score reward following the reward design of RaFe, while keeping all other training settings unchanged (Llama-3.1-8B-Instruct, Contriever retriever). As shown in Table 4, the RaFe-style reward substantially improves over the Default model, confirming that reranker feedback itself provides an effective label-free supervision signal. However, RSDG consistently achieves larger gains, indicating that the rank-position discounting and ideal-order normalization in Eq. (2) more effectively transform reranker scores into retrieval-oriented optimization signals.

Method	Reward	Avg. NDCG@10
Default	–	34.5
RaFe-style	Avg. reranker score	39.9 (+5.4)
RAQE	RSDG	43.6 (+9.1)

Table 4: Comparison between a RaFe-style averaged reranker reward and the proposed RSDG reward under the same training setup.

Effect of Training Stages. Table 5 reports a stage-wise ablation of RAQE, designed to examine the individual contributions of each training stage and their interaction. Applying either stage alone already improves retrieval performance over the Default method, indicating that both supervised distillation (Stage 1) and GRPO-based preference optimization (Stage 2) are individually effective. Notably, Stage 1 provides a strong initialization, while Stage 2 further refines the policy with preference signals. When combined, the two stages exhibit clear synergy and consistently achieve the strongest performance across datasets, demonstrating the effectiveness of the proposed distill-then-optimize framework.

Robustness to Reranker Choice. Since our reward relies on cross-encoder reranker scores⁴—including BGE-Reranker (our default setup), GTE-Reranker, and MXBAI-Reranker—we analyze whether RAQE is sensitive to the specific reranker used to compute the training reward. We train the same policy model with RAQE using three different widely used rerankers. As summarized in Table 6, RAQE consistently improves retrieval performance

Reward Reranker	Default	RAQE (with BGE-Reranker)	RAQE (with MXBAI-Rerank)	RAQE (with GTE-Reranker)
NDCG@10	38.4	45.9 (+7.5)	45.4 (+7.0)	45.5 (+7.1)

Table 6: NDCG@10 of RAQE with three reranker models used for reward computation, showing improvements relative to the Default method. Numbers report average NDCG@10 across eight datasets in Table 1 under sparse retrieval (BM25).

across all three rerankers. This suggests that the proposed RSDG reward formulation in Eq. (2) is robust to the choice of reranker. Among them, we use BGE-Reranker (BAAI/bge-reranker-v2-m3) as the default reranker, as it achieves the best performance.

Additionally, we provide further experimental results, including the results with other retrieval metrics (Recall, MAP, and MRR) in Appendix C.1; the synergy obtained when applying RAQE-trained models to existing training-free QE methods in Appendix C.2; an analysis of reward computation using sparse versus dense retrievers in Appendix D.1; a detailed latency breakdown across QE methods in Appendix E.2; and an analysis of the impact of pseudo-passage generation length in Appendix E.3.

4.3 Downstream Performance

While retrieval metrics such as NDCG@10 measure ranking quality, it is critical to verify whether improvements in retrieval effectiveness translate into actual gains in downstream tasks. Therefore, in this section we evaluate whether the retrieval improvements obtained by RAQE lead to better task-level performance. We consider two representative downstream settings: (1) Retrieval-augmented Generation (RAG) and (2) Fact Verification. ExpandR relies on oracle relevance labels and is therefore included only as a label-supervised reference. In the experiments, we use a fixed retrieval setup: BM25 is used to retrieve the top-5 chunks for each query (or claim). We additionally analyze the sensitivity to different retrieval depths in Appendix C.3.

Retrieval-augmented Generation. For this downstream task, we use the Natural Questions (NQ) corpus from the BEIR benchmark (Thakur et al., 2021) and randomly sample 500 queries from its test split. The retrieved chunks from the corpus are appended to the query, and the final answer is then generated by Llama-3.1-70B-Instruct. We report Exact Match (EM) and F1 scores in Table 7 to assess the correctness of the generated answers when deploying RAG with different QE methods. The results reveal that RAQE yields the largest performance improvement over Default, showing that its superior retrieval gains over other QE methods translate into the best generation accuracy.

Fact Verification. For this task, we use FEVER corpus (Thorne et al., 2018) and sample 500 claims from its test split for fact verification. Given the retrieved chunks, the model (Llama-3.1-70B-Instruct) predicts SUPPORTS, REFUTES, or NOT ENOUGH INFO for each claim. We report Accuracy (Acc) and Macro-F1 in Table 8 to assess the fact verification performance when using different QE methods. The results show that the improved retrieval performance of RAQE leads to meaningful gains in this task, achieving the best overall performance compared to other QE methods.

That is, the improvements of RAQE in retrieval effectiveness translate into stronger downstream performance on both tasks. Additional analyses with different retrieval depths are provided in Appendix C.3, showing that the optimal context budget is task-dependent while the conclusions remain consistent. These results suggest that reranker-aligned training leads to retrieval improvements that generalize well to downstream tasks.

⁴HuggingFace Checkpoints: BGE-Reranker (BAAI/bge-reranker-v2-m3); GTE-Reranker (Alibaba-NLP/gte-reranker-modernbert-base); and MXBAI-Rerank (mixedbread-ai/mxbai-rerank-large-v2).

Method	ExpandR	Default	HyDE	MuGI	Query2Doc	Word2Passage	RAQE
EM	0.322	0.270	0.304	0.252	0.286	0.304	0.316
F1	0.397	0.343	0.387	0.319	0.373	0.393	0.397

Table 7: Open-domain QA performance on Natural Questions (NQ), measured using Exact Match (EM) and F1. Higher is better.

Method	ExpandR	Default	HyDE	MuGI	Query2Doc	Word2Passage	RAQE
Acc	0.566	0.540	0.530	0.546	0.538	0.514	0.582
Macro-F1	0.453	0.438	0.426	0.437	0.435	0.417	0.461

Table 8: Fact verification performance on FEVER, measured using Accuracy (Acc) and Macro-F1. Higher is better.

5 Conclusion

In this paper, we introduced RAQE, a label-free query expansion framework that aligns pseudo-passage generation with retrieval objectives through reranker-derived preference signals. By replacing costly human relevance annotations with the proposed reranker-shaped RSDG reward and optimizing the generator using group-relative policy optimization, RAQE enables effective retrieval-aligned query expansion while maintaining low inference latency through single-pass generation. Extensive experiments across multiple IR benchmarks demonstrate that RAQE consistently improves retrieval quality over strong LLM-based QE baselines and generalizes across both sparse and dense retrieval settings. Furthermore, these retrieval improvements translate into stronger downstream performance in RAG and fact verification tasks. Overall, reranker-aligned policy optimization provides an effective and scalable alternative to supervised QE methods for improving retrieval effectiveness.

Reproducibility Statement

We provide detailed descriptions of model backbones, training procedures, and evaluation protocols to ensure reproducibility. All experiments are conducted using standard benchmarks, including MS MARCO and BEIR datasets, with clearly specified splits and metrics. Implementation details are provided in Appendix A.2, data statistics in Appendix B, efficiency analysis in Appendix E, and model robustness in Appendix F. All prompt templates used in our experiments are provided in Appendix G. The implementation is publicly available at <https://github.com/DISL-Lab/RAQE>.

Acknowledgments. This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2024-00445087, Enhancing AI Model Reliability through Domain-Specific Automated Value Alignment Evaluation). Additionally, this work was supported by the New Faculty Startup Infrastructure Program funded by the Korea government (No. RS-2026-25492133, Intelligent Survey Agents for AI Co-Scientist). This work was also supported by INNOCORE (No. N10260110, AI Meta Scientist).

References

- Abdelrahman Abdallah, Mahmoud Abdalla, Bhawna Piryani, Jamshid Mozafari, Mohammed Ali, and Adam Jatowt. Rerankarena: A unified platform for evaluating retrieval, reranking and rag with human and llm feedback. In *CIKM*, 2025.
- Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Shivani Tiwary, Tong Wang, Michael Wu, and Wen-tau Yu. MS MARCO: A human generated MACHine reading COMprehension dataset. *arXiv preprint arXiv:1611.09268*, 2016.

- Vera Boteva, Demian Gholipour, Artem Sokolov, and Stefan Riezler. A full-text learning to rank dataset for medical information retrieval. In *ECIR*, 2016.
- Claudio Carpineto and Giovanni Romano. A survey of automatic query expansion in information retrieval. *ACM Computing Surveys*, 2012.
- Jeonghwan Choi, Minjeong Ban, Minseok Kim, and Hwanjun Song. Word2passage: Word-level importance re-weighting for query expansion. In *Findings of the ACL*, 2025.
- Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. Precise zero-shot dense retrieval without relevance labels. In *ACL*, 2023.
- Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. Efficiently teaching an effective dense retriever with balanced topic aware sampling. In *SIGIR*, 2021.
- Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *TMLR*, 2022.
- Song Lai, Haohan Zhao, Rong Feng, Changyi Ma, Wenzhuo Liu, Hongbo Zhao, Xi Lin, Dong Yi, Qingfu Zhang, Hongbin Liu, et al. Reinforcement fine-tuning naturally mitigates forgetting in continual post-training. *arXiv preprint arXiv:2507.05386*, 2025.
- Jack Lanchantin, Angelica Chen, Janice Lan, Xian Li, Swarnadeep Saha, Tianlu Wang, Jing Xu, Ping Yu, Weizhe Yuan, Jason E Weston, et al. Bridging offline and online reinforcement learning for llms. *arXiv preprint arXiv:2506.21495*, 2025.
- Victor Lavrenko and W. Bruce Croft. Relevance-based language models. In *SIGIR*, 2001.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, and Sebastian Riedel. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *NeurIPS*, 2020.
- Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. In-batch negatives for knowledge distillation with tightly-coupled teachers for dense retrieval. In *Proceedings of the 6th Workshop on Representation Learning for NLP (ReL4NLP)*, 2021.
- Gabriel Maia, Siegfried Handschuh, and André Freitas. WWW’18 open challenge: Financial opinion mining and question answering. In *WWW*, 2018.
- Shengyu Mao, Yong Jiang, Boli Chen, Xiao Li, Peng Wang, Xinyu Wang, Pengjun Xie, Fei Huang, Huajun Chen, and Ningyu Zhang. RaFe: Ranking feedback improves query rewriting for RAG. In *Findings of the EMNLP*, 2024.
- Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. RocketQA: An optimized training approach to dense passage retrieval for open-domain question answering. In *NAACL*, 2021.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*, 2023.
- Royi Rassin, Yaron Fairstein, Oren Kalinsky, Guy Kushilevitz, Nachshon Cohen, Alexander Libov, and Yoav Goldberg. Evaluating D-MERIT of partial-annotation on information retrieval. In *EMNLP*, 2024.
- Joseph John Rocchio. Relevance feedback in information retrieval. In *The SMART Retrieval System: Experiments in Automatic Document Processing*. Prentice-Hall, 1971.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.
- Hwanjun Song, Jeonghwan Choi, and Minseok Kim. Aligning extraction and generation for robust retrieval-augmented generation. In *WSDM*, 2026.
- Rikiya Takehi, Ellen M. Voorhees, Tetsuya Sakai, and Ian Soboroff. LLM-assisted relevance assessments: When should we ask LLMs for help? In *SIGIR*, 2025.
- Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. *NeurIPS*, 2021.
- James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*, 2018.
- Liang Wang, Nan Yang, and Furu Wei. Query2doc: Query expansion with large language models. In *EMNLP*, 2023.
- Yining Wang, Liwei Wang, Yuanzhi Li, Di He, and Tie-Yan Liu. A theoretical analysis of ndcg type ranking measures. In *COLT*, 2013.
- Zihan Wang, Zihan Liang, Zhou Shao, Yufei Ma, Huangyu Dai, Ben Chen, Lingtao Mao, Chenyi Lei, Yuqing Ding, and Han Li. Infogain-rag: Boosting retrieval-augmented generation through document information gain-based reranking and filtering. In *EMNLP*, 2025.
- Fangyuan Xu, Weijia Shi, and Eunsol Choi. Recomp: Improving retrieval-augmented lms with compression and selective augmentation. *arXiv preprint arXiv:2310.04408*, 2023.
- Zhichao Xu, Shengyao Zhuang, Xueguang Ma, Bingsen Chen, Yijun Tian, Fengran Mo, Jie Cao, and Vivek Srikumar. Rethinking on-policy optimization for query augmentation. *arXiv preprint arXiv:2510.17139*, 2025.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *EMNLP*, 2018.
- Sijia Yao, Pengcheng Huang, Zhenghao Liu, Yu Gu, Yukun Yan, Shi Yu, and Ge Yu. ExpandR: Teaching dense retrievers beyond queries with LLM guidance. *arXiv preprint arXiv:2502.17057*, 2025.
- Le Zhang, Yihong Wu, Qian Yang, and Jian-Yun Nie. Exploring the best practices of query expansion with large language models. In *Findings of the EMNLP*, 2024.

A Details of RAQE

A.1 Optimization

Phase 1: Supervised Distillation. We warm-start the QE policy $\pi_\theta(e \mid q)$ via supervised fine-tuning on teacher-generated query-expansion pairs (q, e^*) . The objective follows a standard sequence-level negative log-likelihood formulation as described in the main text. This stage teaches the model to generate concise, well-formed pseudo-passages and provides a stable initialization for Phase 2.

Phase 2: Reranker-Aligned Group-Relative Policy Optimization (GRPO). Starting from the distilled checkpoint, we optimize π_θ using GRPO. For each query q , we construct a fixed rollout group $\{e_1, \dots, e_g\}$ with $g=8$ candidates.

For each candidate e_i , we form an expanded query $q'_i = \text{concat}(q, e_i)$, and compute the reward $r_i = \Delta(q'_i, q)$ using cached retrieval results and reranker scores.

Given within-group rewards, we compute standardized group-relative advantages and optimize the policy using the GRPO objective (see Eq. 5 in the main text).

A.2 Implementation

Training configuration. We train RAQE with a two-stage distill-then-optimize procedure. Stage 1 performs supervised fine-tuning for 5 epochs. Stage 2 applies reranker-aligned GRPO for 2 epochs with group size $g=8$ and reward depth $k=30$. The clipped surrogate objective uses $\epsilon=0.2$. Unless stated otherwise, hyperparameters are shared across datasets.

Optimization and hardware. Training uses the AdamW optimizer with learning rate 5×10^{-6} and batch size 8. We train using LoRA-based fine-tuning with bfloat16 precision on two NVIDIA RTX PRO 6000 GPUs.

Stage 1 (supervised distillation) is trained for 5 epochs over 1,010 update steps, taking about 12 minutes. Stage 2 (reranker-aligned GRPO) is trained for 2 epochs over 1,730 update steps, taking about 24 minutes, with group size $g=8$, reward depth $k=30$, and clipping parameter $\epsilon=0.2$. The larger number of update steps is due to rollout-based training, where each query is associated with multiple candidates. Since rollout candidates, retrieval results, and reranker scores are precomputed offline, training does not require additional retriever or reranker calls, enabling efficient and stable optimization. Unless otherwise specified, all reported results are obtained from a single training run per configuration.

Inference. At inference time, RAQE generates a single concise pseudo-passage per query. For sparse retrieval, we concatenate the pseudo-passage with repeated instances of the original query (five times), following common practice in QE. For dense retrieval, we average the encoder representations of the original query and the pseudo-passage to form the expanded query embedding. No reranking or iterative generation is performed at test time.

B Data Statistics

We construct training data from five benchmarks with official training splits: NFCorpus, FiQA, FEVER, HotpotQA, and MS MARCO. For each training query q , we generate a teacher pseudo-passage e^* and form the expanded query $q' = \text{concat}(q, e^*)$.

Query sampling. From each dataset, we target 3,000 randomly sampled training queries to construct an initial pool for distillation and preference optimization.

Filtering by retrieval impact. We apply a query-level filtering step to retain only cases where query expansion is beneficial. Specifically, for each query q , we compute the reranker-shaped gain $\Delta(q', q)$ defined in Eq. (3) and keep examples satisfying $\Delta(q', q) > 0$. After

Method	R@50	MAP@10	MRR@10
Label-supervised reference	0.6234	0.3613	0.5460
Default	0.5662	0.2929	0.4791
HyDE	0.5918	0.3239	0.5058
Query2Doc	0.5959	0.3301	0.5122
MuGI	0.5413	0.2870	0.4583
Word2Passage	0.5190	0.2732	0.5543
RAQE (ours)	0.6265	0.3574	0.5412

Table 9: Additional retrieval metrics with BM25. The label-supervised ExpandR reference uses relevance supervision during training.

Dataset	Sparse (BM25)			Dense (Contriever)		
	MuGI	MuGI + Ours	Ours	MuGI	MuGI + Ours	Ours
NFCorpus	28.5	35.9	36.0	22.0	28.8	33.5
FiQA	20.2	20.7	24.5	15.2	18.6	28.8
FEVER	81.4	83.9	84.5	47.8	71.3	80.9
HotpotQA	48.7	68.2	69.1	22.4	45.0	62.9
MS MARCO	51.4	58.8	65.5	51.3	54.0	56.9
ArguAna	27.6	28.3	28.3	13.5	16.5	28.5
NQ	31.3	45.1	44.4	19.8	35.2	40.9
SciDocs	13.1	14.2	14.7	9.3	11.7	16.4

Table 10: Applying RAQE on top of MuGI under both sparse and dense retrieval (NDCG@10). “MuGI + Ours” denotes applying RAQE to MuGI-generated expansions. We additionally report the standalone performance of RAQE (Ours). Our method consistently improves MuGI while achieving strong standalone performance, demonstrating that it is complementary to existing multi-generation-based QE strategies.

filtering, the number of remaining queries is: NFCorpus (753), FiQA (837), FEVER (2,103), HotpotQA (1,882), and MS MARCO (878).

Overall, this process yields 6,453 training queries, which are shared by both Stage 1 distillation and Stage 2 GRPO.

C Additional Evaluation Results

C.1 Additional Retrieval Metrics

We additionally report Recall@50, MAP@10, and MRR@10, which complement NDCG@10 by measuring retrieval coverage (R@50), overall precision (MAP@10), and early-rank accuracy (MRR@10). Results are macro-averaged over the eight datasets. The macro-averaged results are summarized in Table 9.

C.2 Synergy between RAQE and Training-Free QE Methods

We investigate whether our method is complementary to existing query expansion strategies by applying it on top of MuGI (Zhang et al., 2024), a representative multi-generation QE method that performs strongly across both sparse and dense retrieval settings. We denote this combination as “MuGI + Ours”, where RAQE is applied on top of MuGI-generated expansions. Table 10 shows that our method consistently improves MuGI across all datasets in both retrieval paradigms. In addition, RAQE alone already achieves strong performance, often outperforming MuGI, while further gains can be obtained when combining the two. In the sparse setting, we observe gains on every dataset, with particularly large improvements on HotpotQA (+19.5) and NQ (+13.8). Similar trends are observed in the dense setting, where consistent improvements are also achieved. These results demonstrate that reranker-aligned optimization is orthogonal to existing QE methods and can be combined with them to further improve retrieval performance across different retrievers.

Dataset / Metric	$k=1$	$k=3$	$k=5$ (Orig.)
NQ EM	0.270	0.304	0.316
NQ F1	0.357	0.382	0.397
FEVER Acc.	0.602	0.604	0.582
FEVER Macro-F1	0.476	0.475	0.461

Table 11: Sensitivity of downstream performance to retrieval depth (k). We vary the number of retrieved passages while keeping all other experimental settings unchanged.

Training Reward	Test Sparse	Test Dense
Sparse reward	45.9	40.5
Dense reward	45.3	43.6

Table 12: Cross-evaluation by reward source.

C.3 Sensitivity to Retrieval Depth

We additionally evaluate the effect of retrieval depth ($k=1,3,5$) on the downstream tasks while keeping all other settings unchanged. For NQ, larger retrieval depths consistently improve EM and F1, suggesting that additional retrieved evidence benefits answer generation. For FEVER, $k=1$ or $k=3$ slightly outperform $k=5$, indicating that additional retrieved passages may introduce distracting evidence for verification. These results suggest that the optimal retrieval depth depends on the downstream task, while our choice of $k=5$ provides a balanced evaluation setting.

D Additional Reward Analyses

D.1 Reward Source: Sparse vs. Dense

We analyze how the retrieval source used to compute the reward affects downstream performance. We train separate models using rewards computed from sparse (BM25) and dense (Contriever) retrieval, and evaluate them under both sparse and dense test settings. All results are averaged across datasets using the same backbone (Llama-3.1-8B-Instruct) as in the main experiments. The cross-evaluation results are reported in Table 12.

Overall, aligning the reward source with the target retriever yields the strongest performance, while cross-retrieval generalization remains competitive.

D.2 Sensitivity to Reward Depth

We analyze the effect of the reward depth used in RSDG computation by varying the cutoff used to compute the reward ($k \in \{10, 30, 50\}$), while keeping all other training settings unchanged. Table 13 reports the resulting average NDCG@10 across the eight datasets under sparse retrieval. The results show that RAQE is not particularly sensitive to the reward depth, with only marginal performance differences across different values of k . This suggests that the proposed reward formulation is robust to the choice of reward depth, and that the default setting of $k = 30$ provides stable performance.

E Efficiency Analyses

We provide a detailed latency breakdown and analyze generation-length sensitivity.

E.1 Effect of Rollout Group Size

We analyze the effect of the rollout group size $g \in \{8, 4, 2\}$ on retrieval performance. For smaller g , we subsample candidates from the same precomputed rollout pool using a fixed rule, while keeping all other training settings unchanged. As shown in Table 14, reducing the

Reward Depth (k)	10	30 (Default)	50
Avg. NDCG@10	46.6	45.9	46.4

Table 13: Sensitivity of RAQE to the reward depth used in RSDG computation. All models are evaluated using NDCG@10 under the same setup as Table 1.

g	NFC	FiQA	FEV	HP	MSM	Avg
8	36.0	24.5	84.5	69.1	65.5	55.9
4	35.8	23.6	84.6	69.7	64.4	55.6
2	36.4	24.4	83.7	68.8	62.6	55.2

Table 14: Effect of rollout group size g on retrieval performance (NDCG@10). Results are reported on five in-domain datasets. Smaller g values are obtained by subsampling from the same rollout pool. While performance remains stable from $g = 8$ to $g = 4$, further reducing to $g = 2$ leads to consistent degradation on average.

rollout size from $g = 8$ to $g = 4$ results in only minor performance degradation, indicating that the method is relatively robust to moderate reductions in group size. However, further reducing to $g = 2$ consistently decreases performance on average. This suggests that larger rollout groups provide richer relative preference signals, which are beneficial for group-relative policy optimization. Notably, when $g = 2$, the training effectively reduces to pairwise comparisons, limiting the diversity of relative comparisons within each query. This result highlights the importance of group-level preference structure beyond simple pairwise supervision. Although $g = 4$ offers a favorable trade-off between performance and computational cost, $g = 8$ consistently achieves the best average performance and is therefore used as the default setting.

E.2 Latency Breakdown

Table 15 provides a detailed breakdown of end-to-end latency per query, decomposed into generation and retrieval components for each method. Latency is measured as the average per-query inference time, macro-averaged over all eight evaluation datasets.

The Default method, which does not perform query expansion, is included as a latency reference rather than a direct comparison target.

We group methods into *single-generation* and *multi-generation* approaches. Single-generation methods (e.g., Query2Doc and RAQE) produce a single pseudo-passage per query, while multi-generation methods (e.g., HyDE, MuGI, and Word2Passage) generate multiple candidates, incurring higher generation cost.

Within each group, RAQE achieves competitive latency while maintaining a single-pass design, whereas multi-generation methods exhibit significantly higher latency due to repeated decoding.

E.3 Impact of Generation Length

As shown in Table 16, we analyze the impact of the maximum generation length on retrieval effectiveness. Performance improves as the expansion length increases up to 128 tokens, but degrades with longer generations, suggesting that intermediate-length expansions strike a better balance between informativeness and noise.

F Model Scaling and Robustness

We evaluate scaling behavior across generator sizes and robustness across generator families.

Method	Gen. (s)	Ret. (s)	Total (s)
Default	0.00	0.15	0.15
<i>Single-generation methods</i>			
Query2Doc (4-shot)	2.90	0.05	2.95
RAQE (ours)	<u>3.29</u>	<u>0.17</u>	<u>3.45</u>
<i>Multi-generation methods</i>			
HyDE ($\times 8$)	12.51	0.36	12.87
MuGI ($\times 5$)	<u>12.54</u>	<u>0.33</u>	12.87
Word2Passage ($\times 5$)	13.14	0.09	13.23

Table 15: Latency breakdown per query (seconds), macro-averaged over 8 datasets. Prompting overhead is negligible and thus included in the generation time. Methods are grouped into single-generation and multi-generation approaches. **Bold** indicates the fastest method within each group, and underlining indicates the second-fastest.

Max Tokens	32	64	128 (Default)	256
Sparse Avg. (NDCG@10)	52.7	54.4	55.9	53.3

Table 16: Effect of generation length on retrieval performance (Sparse NDCG@10). Results are averaged over the five training datasets (NFCorpus, FiQA, FEVER, HotpotQA, and MS MARCO), hence values are not directly comparable to the eight-dataset averages in the main tables.

Method	Default	HyDE	MuGI	Query2 Doc	Word2 Passage	RAQE (3B)
Sparse (Avg)	38.4	42.5	35.1	39.6	43.8	44.3

Table 17: Sparse retrieval performance with a 3B generator (Llama-3.2-3B-Instruct).

F.1 Performance across Model Sizes

Table 17 reports sparse retrieval performance averaged across datasets when using a smaller 3B generator, allowing us to examine how RAQE behaves under reduced model capacity. While larger generators generally improve retrieval performance, retrieval gains do not always scale monotonically with model size. Similar observations have also been reported for prompting-based query expansion methods such as Word2Passage (Choi et al., 2025), where larger backbones do not consistently outperform smaller ones across all datasets. These findings suggest that retrieval effectiveness depends not only on generator scale but also on how query expansions are generated.

Despite the reduced model capacity, RAQE maintains the strongest average performance among all methods.

F.2 Robustness across Generator Architectures (Qwen)

We evaluate RAQE on a different generator family to assess architectural robustness. We train RAQE using Qwen3-4B-Instruct and compare it against label-free prompting baselines using Qwen3-30B-A3B-Instruct.

Discussion. With a substantially smaller 4B Qwen generator, RAQE achieves the highest macro-average NDCG@10 among label-free methods and matches or outperforms prompting-based baselines that rely on 30B generators on most in-domain datasets. These results, summarized in Table 18, indicate that reranker-aligned optimization generalizes across generator families and does not rely on generator scale alone.

Method	Gen.	NFC	FiQA	FEV	HP	MSM	Arg	NQ	SciD	Avg
Default	–	31.8	24.2	68.6	45.4	42.6	26.2	22.5	14.5	34.5
HyDE	4B	34.2	19.5	79.1	59.8	56.6	27.3	40.8	13.3	41.3
	30B	35.5	20.6	80.8	64.2	59.3	26.2	45.7	13.8	43.3
Query2Doc	4B	34.7	25.3	77.9	64.3	61.1	27.1	38.0	14.1	42.8
	30B	34.2	25.2	74.0	66.3	62.5	27.5	39.2	14.2	42.9
MuGI	4B	32.1	21.2	74.7	61.9	58.9	27.4	39.7	12.5	41.1
	30B	35.2	23.1	79.8	67.7	62.8	27.3	45.4	13.2	44.3
Word2Passage	4B	21.2	19.3	73.1	63.6	52.3	6.8	33.4	6.1	34.5
	30B	29.5	21.9	76.5	69.1	50.2	29.7	39.0	14.8	41.3
RAQE (ours)	4B	35.5	25.4	81.6	66.0	63.1	27.8	40.4	14.3	44.3

Table 18: Sparse retrieval results (NDCG@10) with Qwen generators. Baselines are evaluated using both 4B and 30B generators when available.

Seed	NFC	FiQA	FEV	HP	MSM	Arg	NQ	SciD	Avg
42 (submitted)	36.0	24.5	84.5	69.1	65.5	28.3	44.4	14.7	45.9
123	35.7	24.8	84.3	70.8	64.8	28.1	45.0	14.6	46.0
3407	35.8	24.1	84.2	70.7	66.1	28.3	45.8	14.8	46.2

Table 19: Performance of RAQE across different random seeds on the eight BEIR datasets. Results are reported using NDCG@10.

F.3 Robustness across Random Seeds

To evaluate the robustness of RAQE to random initialization, we repeat training using three random seeds (42, 123, and 3407) while keeping all other training settings unchanged. Table 19 reports the macro-average NDCG@10 across the eight BEIR datasets. The results show only minor variation across different random seeds. Notably, the submitted result (Seed 42) is not a favorable outlier, as the remaining seeds achieve comparable or slightly better performance. These findings indicate that the improvements of RAQE are stable and are not driven by a particular random seed.

G Prompt Templates

We summarize the prompt templates used for RAQE and the baselines. For baseline methods, we follow the prompts from the original papers or official implementations as closely as possible, and only omit long in-context demonstrations when necessary for brevity.

G.1 RAQE (Ours)

We use a single-pass prompt that encourages one concise pseudo-passage while avoiding meta-commentary:

Instruction: Write a single cohesive paragraph (at least three sentences) that directly addresses the query. Include concrete details when possible, avoid meta commentary, and return only the passage text.
Query: {query}

G.2 Baselines

HyDE (Gao et al., 2023). We follow the dataset-specific instructions from the original HyDE paper. For the datasets used in our experiments, the prompts are:

- **General / Web Search-style datasets (e.g., NFCorpus, NQ, MS MARCO, HotpotQA):**

Please write a passage to answer the question
Question: {query}
Passage:

- **FiQA:**

Please write a financial article passage to answer the question
Question: {query}
Passage:

- **SciDocs / scientific retrieval:**

Please write a scientific paper passage to answer the question
Question: {query}
Passage:

- **ArguAna:**

Please write a counter argument for the passage
Passage: {query}
Counter Argument:

- **FEVER / claim verification style:**

Please write a passage to verify the claim
Claim: {query}
Passage:

Query2Doc (Wang et al., 2023). We follow the official 4-shot in-context prompt from Query2Doc, using the released dataset-specific demonstrations. For brevity, we omit the full demonstration contents and show only the prompt structure:

```
Write a passage that answers the given query:
Query: <example query 1>
Passage: <example passage 1>
Query: <example query 2>
Passage: <example passage 2>
Query: <example query 3>
Passage: <example passage 3>
Query: <example query 4>
Passage: <example passage 4>
Query: {query}
Passage:
```

MuGI (Zhang et al., 2024). We use the released MuGI passage-generation prompt:

```
You are PassageGenGPT, an AI capable of generating concise, informative,
and clear pseudo passages on specific topics. Generate one passage that is
relevant to the following query: '{query}'. The passage should be concise,
informative, and clear.
```

Word2Passage (Choi et al., 2025). We follow the original prompt format, which asks the model to generate passage-, sentence-, and word-level outputs in JSON:

```
Generate a passage, a sentence, and words that answer the given QUERY.
Terms that are important for answering the QUERY should frequently appear
in the generation of the passage, the sentence, and words.
### Definition:
**passage**: Answer the given QUERY in a passage perspective by generating
an informative and clear passage.
**sentence**: Answer the given QUERY in a sentence perspective by
generating a knowledge-intensive sentence.
**word**: Answer the given QUERY in a word perspective by generating a
list of words.
### QUERY:
{query}
### FINAL OUTPUT JSON FORMAT (strictly follow this structure):
```

```
{  
  "passage": "Your passage here",  
  "sentence": "Your sentence here",  
  "word": [Your words here]  
}
```

(From here on, only produce the final output in the specified JSON format.)

ExpandR (Yao et al., 2025). ExpandR considers multiple expansion strategies (e.g., query rewriting, keyword generation, chain-of-thought, and passage generation). In our experiments, we use the passage generation variant, which is directly comparable to other pseudo-passage-based QE methods:

```
Please write a passage to answer the question:  
Question: {query}  
Passage:
```