Reinforced Query Reasoners for Reasoning-intensive Retrieval Tasks

Anonymous ACL submission

Abstract

Traditional information retrieval (IR) methods excel at textual and semantic matching but struggle in reasoning-intensive retrieval tasks that require multi-hop inference or complex semantic understanding between queries and documents. One promising solution is to explicitly rewrite or augment queries using large language models (LLMs) to elicit reasoning-relevant content prior to retrieval. However, the widespread use of large-scale LLMs like GPT-4 or LLaMA3-70B remains impractical due to their high inference cost and limited deployability in real-world systems. In this work, we introduce **Reinforced Query Reasoner (RQR)**, a family of small-scale language models for query reasoning and rewriting in reasoning-intensive retrieval. Our approach frames query reformulation as a reinforcement learning problem and employs a novel semi-rule-based reward function. This enables smaller language models, *e.g.*, `Qwen2.5-7B-Instruct` and `Qwen2.5-1.5B-Instruct`, to achieve reasoning performance rivaling large-scale LLMs without their prohibitive inference costs. Experiment results on BRIGHT (Su et al., 2024) benchmark show that, with BM25 as retrievers, both `RQR-7B` and `RQR-1.5B` models significantly outperform existing baselines, including prompt-based query reasoners and some latest dense retrievers trained for reasoning-intensive retrieval tasks, offering superior adaptability for real-world deployment. All code, models and dataset will be publicly released.

1 Introduction

The Information retrieval system (IR) (Zhu et al., 2023) plays a critical role in satisfying information needs, enabling users to locate relevant materials from vast repositories of documents, Web pages, and structured records. While existing retrieval methods — including text matching (Robertson and Zaragoza, 2009) and semantic representation techniques (Devlin, 2018; Liu, 2019; Chen et al., 2024a; Ma et al., 2024) — have achieved considerable success, they often fall short in scenarios demanding intensive reasoning and modeling the implicit correlations between documents and user queries (Su et al., 2024), where the problem is even more severe for recent retrieval-augmented generation (RAG) applications (Zhao et al., 2024). In real-world scenarios and tasks, user queries are frequently complex and underspecified, requiring intensive reasoning to uncover latent relationships that transcend surface-level semantic or textual similarity. For example, an economist may seek articles that apply the same economic theory to different cases, or a programmer may need to find an alternative function with the same implementation logic.

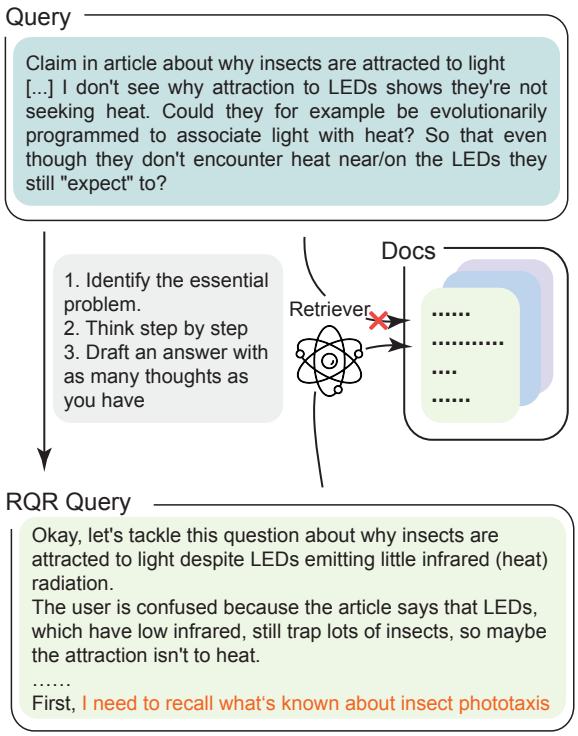

Figure 1: An example query reasoning with LLM. The query is sampled from the Biology Subtask of BRIGHT (Su et al., 2024) benchmark.

We refer to such task as **reasoning-intensive retrieval** (Shao et al., 2025), which has been proved to be challenging with poor performance.

To address this issue, two research directions have been proposed. One is to train novel retriever or reranker models (Shao et al., 2025; Weller et al., 2025), training on task-specific reasoning data. The other is to apply **query reasoning and rewriting** to the given query (Su et al., 2024; Niu et al., 2024), leveraging the frontier reasoning capabilities of large language models (LLMs) with chain-of-thought reasoning (Wei et al., 2022) to generate an intermediate reasoning result as **reasoned query**, which instead will be used to retrieve the relevant documents. Figure 1 shows an example of query reasoning. Existing query reasoning approaches mainly rely on large-scale LLMs (*e.g.*, GPT-4o (OpenAI et al., 2024) or LLama3-70B (Grattafiori et al., 2024)) with a high computational and deployment cost. Although they can be leveraged to optimize the problem with integrated reasoning and retrieval process, the inference latency of such LLMs hinders the applications and deployment on real-time or interactive retrieval.

In this paper, we introduce RQR, a family of small-scale language models for query reasoning and rewriting. To the best of our knowledge, this is the first model family specifically trained for query reasoning in reasoning-intensive retrieval tasks. Inspired by previous works using Reinforcement Learning with Verifiable Rewards (RLVR) to enhance LLMs' reasoning (Guo et al., 2025; Qwen et al., 2025), we developed a novel semi-rule-based reward function for GRPO(Group Relative Policy Optimization) (Shao et al., 2024; Guo et al., 2025), enabling reinforcement learning on the query reasoning of smaller language models. Beyond that, we further propose an automatic data curation pipeline for training reasoning-based rewriting with public available dataset. Experiment results on BRIGHT (Su et al., 2024) benchmark show that our model can achieve the ndcg@10 metric at 27.9, outperforming the metric of GPT-4o at 26.5. This metric is comparable to some large-scale reasoning models, e.g., o1-preview[1], DeepSeek R1(Guo et al., 2025), and QwQ-32B (Qwen et al., 2025), with significantly lower cost of inference. Besides, our proposed models can also work with reasoning-intensive retrievers (Shao et al., 2025) to achieve the best performance. This demonstrates that our models possess strong flexibility to adapt to different retrieval pipelines.

In summary, our **main contributions** are listed as follows:

- **Query rewriting models for reasoning-intensive tasks:** We propose RQR family (7B and 1.5B) specifically trained for query reasoning and rewriting in reasoning-intensive retrieval tasks. Our models with extreme smaller parameters are comparable to state-of-art large reasoning models such as GPT-4o on specific tasks with significantly reduced computational cost and retrieval latency. Our query reasoning and rewriting models is proved to be generalized to various tasks and jointly applied to different existing retrievers and rerankers to achieve better performance.
- **Semi-rule-based reward function for RL:** The reward function inherits the advantage of existing functions based on semantic similarity, which evaluates the relevance enhancement between queries and retrieval documents. It offers a range of advantages, including strong robustness, high computational efficiency in avoid of reward hacking.
- **Automatic data curation pipeline:** The data curation pipeline proposed in this paper is specifically designed to build training data for query rewriting tasks. It optimizes training without the need of large-scale supervised rewrite data, which are often unavailable in real applications and scenarios.

2 Related works

Reasoning-intensive Retrieval In recent years, dense retrieval has achieved remarkable progress in retrieval accuracy, propelled by the rapid evolution of foundation models and innovative training methodologies. Nowadays, BERT (Devlin, 2018)-based and LLM-based (Wang et al., 2023) embedding models have been widely used in multiple retrieval tasks, achieving great success as general-purpose retrievers (Wang et al., 2022; Li et al., 2023; Chen et al., 2024a; Khattab and Zaharia, 2020). However, previous works (Su et al., 2024) have demonstrated that most of those existing BERT-based or LLM-based retrievers and re-rankers cannot handle the task of reasoning-intensive retrieval. Most of those sparse or dense retrievers perform poorly on BRIGHT

[1]https://openai.com/index/introducing-openai-o1-preview/

benchmark[2]. These results indicate that the tasks of reasoning-intensive retrieval should be handled with reasoning-enhanced models specifically. Some researchers tried to train reasoning-enhanced retrievers (Shao et al., 2025) or rerankers (Weller et al., 2025) with public or LLM-Synthesized datasets. Another way is to apply LLMs for query reasoning and rewriting. The LLMs take the original queries as input to generate Chain-of-Thought reasoning steps as pseudo queries. The pseudo queries will be issued to the retrievers instead of the original queries. These two approaches are orthogonal and can be combined synergistically. To the best of our knowledge, most of those existing query reasoning approaches (Su et al., 2024; Niu et al., 2024) are based on prompting large-scale LLMs, *e.g.*, GPT-4o (OpenAI et al., 2024) or LLama3-70B (Grattafiori et al., 2024), which is too expensive and time-consuming. To the best of our knowledge, none of those previous works focus on training a small-scaled language model for query reasoning and rewriting tasks.

Reasoning Enhanced by Reinforcement Learning Large reasoning models, *e.g.*, `OpenAI o1`, `Gemini Flash-Thinking`[3], `DeepSeek-R1` (Guo et al., 2025) and `QwQ-32B` (Qwen et al., 2025), have achieved great success in reasoning-intensive areas like coding and mathematical proofs. These models adopt a "slow-thinking" (Wu et al., 2024; Chen et al., 2024b) approach when handling reasoning-intensive tasks: the models will first output a sequence of thinking processes with the tags of "<think></think>" before providing the actual answer. This method has allowed LLMs to enhance reasoning capabilities. Based on the technical report released by DeepSeek (Guo et al., 2025), researchers (Face, 2025; Xie et al., 2025) have tried to reproduce the slow-thinking ability on smaller-scaled LLMs via reinforcement learning based on GRPO (Group Relative Policy Optimization) (Shao et al., 2024) and rule-based reward functions. Compared with process reward models (PRM), the rule-based reward functions have the advantages of being simple and effective, making the model training process easier to scale up. Besides, the rule-based reward functions only focus on the correctness of output results, ignoring the intermediate process, making it immune to reward hacking and increasing the robustness of model training. Moreover, unlike supervised fine-tuning (SFT), reinforcement learning based methods do not force the model to fit every generated token, thereby yielding superior generalization capabilities.

[2]https://brightbenchmark.github.io/

[3]https://deepmind.google/technologies/gemini/flash-thinking/

3 Reinforced Query Reasoner

3.1 Task formulation

Given a query q and a set of candidate documents $D = \{d_1, ..., d_n\}$, the objective is to identify and retrieve a subset of relevant documents from D: $D^+ = \{d_1^+, ..., d_i^+, ..., d_m^+\}$, where $m << n$ leveraging a retriever $\mathcal{RT}$. In the scenario of reasoning-intensive retrieval, we leverage a large language model $\mathcal{LLM}$ to generate the rewritten query $q^{'}$ after query reasoning based on q. $\mathcal{RT}$ will later use $q^{'}$ to retrieve the documents relevant to q. The processes mentioned above can be described with the following equations:

$$q^{'} = \mathcal{LLM}(\text{Inst}; q), \qquad D^+ = \mathcal{RT}(q^{'})$$

where Inst denotes the instructions for query reasoning and rewriting.

3.2 Reinforcement Learning with Semi-Rule-Based Reward

Preliminary Inspired by previous works of large reasoning models *e.g.*, DeepSeek R1 (Guo et al., 2025), we employ the GRPO-based reinforcement learning algorithm to train the LLMs for query reasoning, where the model takes the given query q as input and generates a reasoned query $q^{'}$. The GRPO objective is defined as:

$$\mathcal{L}_{\text{GRPO}}(\theta) = \mathbb{E}_{(q,a)\sim\pi_\theta}\big[w_g \cdot \min\big(r_\theta(q,q^{'}) \cdot \hat{A}(q,q^{'}), \text{clip}(r_\theta(q,q^{'}), 1-\epsilon, 1+\epsilon) \cdot \hat{A}(q,q^{'})\big)\big]$$

Here, $r_\theta(q,q^{'}) = \frac{\pi_\theta(q^{'}|q)}{\pi_{\theta_{\text{old}}}(q^{'}|q)}$ is the importance ratio between the current and reference policy. The advantage function $\hat{A}(q,q^{'})$ is computed based on the group-normalized reward:

$$\hat{A}(q,q^{'}) = \frac{R(q,q^{'}) - \mu_g}{\sigma_g + \delta}$$

where $R(q,q^{'})$ is the reward assigned to the reasoned query $q^{'}$, μ_g and σ_g denote the mean and standard deviation of rewards within the group g, and δ is a small constant to avoid division by zero. The weight w_g optionally re-scales the advantage based on group-level reward variance. This formulation stabilizes training when rewards are sparse or highly variable across different query groups.

Limitations for Previous Rule-based Reward Function Previous approaches (Jiang et al., 2025) of rule-based reward for retrieval tasks are usually calculated based on retrieval evaluation metrics like Recall@K. The metric-based reward function requires both annotated training data and an existing large-scale document collection to serve as the retrieval source, which is difficult to access in reasoning-intensive retrieval tasks.

Semi-Rule-Based Reward for Query Reasoning In this work, we introduce a reward function to evaluate the incremental relevance score from $< q, D^+ >$ to $< q^{'}, D^+ >$. For an reasoning-intensive task, the goal of query reasoning and rewriting is to improve the retrieval performance using reasoned query $q^{'}$ with higher relevance score compared to q. Since the relevance score is computed via an existing relevance model, the reward function is defined as "semi-rule-based reward function".

Each training sample consists of $< q, D^+ >$, where D^+ indicates single or multiple positive documents for q. We define $score_q$ as the sum of the relevance scores between q and each positive document in D^+:

$$score_q = \Sigma_{i \in D^+} \text{Rel}(q, d_i^+)$$

where $Rel(q, d_i^+)$ denotes the relevance score between q and d_i^+ computed via a relevance model. Here we use a pretrained embedding model to encode queries and documents into embeddings, with the cosine similarities as relevance scores. The parameters of the relevance model will not be updated during the model training process. Similarly, the score of the reasoned query $score_{q'}$ is also computed as:

$$score_{q'} = \Sigma_{i \in D^+} \text{Rel}(q^{'}, d_i^+)$$

The overall reward is defined as the average relevance score increment from q to $q^{'}$ of each positive document:

$$R(q, q^{'}) = \frac{score_{q'} - score_q}{|D^+|}$$

Our semi-rule-based reward function inherits a few advantages from the existing rule-based rewards as below: Firstly, the function depicts the semantic relevance based on the existing embedding model like bge-base-en (Chen et al., 2024a), which has been proved to exhibit good performance with robustness and low computational cost. Secondly, unlike the process reward models (PRMs), our method does not rely on intermediate processes supervision, and is therefore inherently immune to reward hacking. These properties collectively contribute to the high computational efficiency and robustness of our method, enhancing its tolerance to noise present in the training data.

3.3 Training Data Curation

Existing training datasets like *e.g.*, MS-MACRO (Bajaj et al., 2018) are helpful for semantic-based retrieval tasks, which are not specifically designed for reasoning-intensive retrieval. Inspired by the data construction process in benchmark BRIGHT (Su et al., 2024), we use the publicly available H4 Stack Exchange Preferences (Lambert et al., 2023) dataset to construct our training data. The dataset contains questions and answers from the Stack Overflow Data Dump for the purpose of preference model training. Each question in the dataset includes at least two answers, and each answer is labeled "is_selected" or not, indicating if the answer is selected and marked as useful by the real users who issued the question. We select QAs with texts only for data curation.

Here are two ways we further obtain the rewritten queries as the "supervision" for query reasoning training:

(1) Given a query for reasoning, a large reasoning model, *e.g.*, QwQ-32B or DeepSeek-R1 is asked to generate the rewritten query based on Chain-of-Thought(CoT) reasoning. The curated data is denoted as **V1-R1** and **V1-QwQ**.

(2) For each question, we use the answer with "selected" tag as the reasoned query from StackExchange by real users, which is denoted as **V2**. Notice that not every question includes a selected answer.

4 Experiment

4.1 Experiment setting

4.1.1 Dataset and metrics

Training We employ two types of the constructed data mentioned in Section 3.3 for training: **V1-R1**, **V1-QwQ** and **V2**. For V2, we use the user-selected answers since the size of V2 is too large to afford the inference cost of large reasoning models. More details can be found in Appendix B.

	StackExchange							Coding		Theorem-based			Avg
	Bio.	Earth.	Econ.	Psy.	Rob.	Stack.	Sus.	Leet.	Pony	AoPS	TheoQ.	TheoT.	
Retrievers with Original Queries													
BM25	18.9	27.2	14.9	12.5	13.6	18.4	15.0	24.4	7.9	6.2	10.4	4.9	14.5
BGE	11.7	24.6	16.6	17.5	11.7	10.8	13.3	26.7	5.7	6.0	13.0	6.9	13.7
ReasonIR	26.2	31.4	23.3	30.0	18.0	23.9	20.5	35.0	10.5	14.7	31.9	27.2	24.4
Seed1.5-Embedding	34.8	46.9	23.4	31.6	19.1	25.4	21.0	43.2	4.9	12.2	33.3	30.5	27.2
Query Reasoner with BM25													
GPT-4o	53.6	53.6	24.3	38.6	18.8	22.7	25.9	19.3	17.7	3.9	18.9	20.2	26.5
Doubao	54.8	53.3	23.7	37.2	22.2	28.1	25.0	21.2	16.4	7.8	21.8	22.7	27.8
Deepseek-V3	56.6	54.2	25.8	38.8	19.9	26.7	26.4	19.8	15.1	6.7	22.5	20.7	27.8
o1-mini	60.2	57.4	24.7	39.3	23.3	26.4	25.4	23.5	13.4	6.9	22.8	16.5	28.3
o1-preview	**64.2**	57.9	27.6	**43.1**	**25.6**	29.1	28.0	21.2	15.9	5.6	24.0	20.5	30.2
Deepseek-R1	62.7	**58.3**	26.0	42.9	21.8	28.1	**30.3**	19.6	10.7	6.0	25.8	22.4	29.6
R1-distill-qwen-7B	33.9	41.6	19.9	31.8	15.1	18.8	16.4	19.7	10.7	6.8	24.5	22.2	21.8
R1-distill-qwen-32B	50.6	49.9	22.9	38.1	20.3	24.6	19.2	19.5	11.3	5.6	24.2	20.2	25.5
QwQ-32B	57.5	56.3	29.9	41.8	19.2	25.7	27.2	21.5	12.8	6.5	25.4	22.8	28.9
RQR-1.5B	46.0	47.1	21.1	31.2	19.8	21.7	24.3	22.5	**21.7**	4.3	19.7	15.9	24.6
RQR-7B	57.9	50.9	21.9	37.0	21.3	27.0	25.6	23.6	14.4	7.0	26.1	22.0	27.9
Query Reasoner with ReasonIR													
LLama3.1-8B-Instruct	37.8	39.6	29.6	35.3	24.1	**31.1**	27.4	28.8	14.5	9.2	26.6	32.3	28.0
GPT-4	43.6	42.9	**32.7**	38.8	20.9	25.8	27.5	31.5	19.6	7.4	33.1	35.7	29.9
RQR-1.5B	36.4	41.1	29.9	34.0	25.2	30.7	25.6	**33.3**	16.8	9.7	35.7	32.7	29.3
RQR-7B	46.2	45.1	31.2	39.6	25.3	28.7	28.4	31.2	16.3	**10.8**	**40.0**	**39.3**	**31.9**

Table 1: Performance comparison on BRIGHT. The best score is shown in bold and the second best is underlined.

Evaluation We use BRIGHT (Su et al., 2024), a novel benchmark for reasoning-intensive retrieval that aims to evaluate the ability of retrieval models to handle complex queries that require deep reasoning. It consists of 1,384 real-world queries from diverse domains with 12 sub-tasks. We adopt the metric **nDCG@10** for the following evaluations.

4.1.2 Baselines

The baselines in our experiments can be divided into these three categories:

Retrievers with Original Queries There are two types of baselines: 1) Traditional baselines in IR systems like BM25 (Robertson and Zaragoza, 2009) for sparse retrieval and bge-large-en (Chen et al., 2024a)for dense retrieval; 2) Reasoning-intensive retrievers like ReasonIR (Shao et al., 2025) and Seed 1.5-Embedding[4]. We keep the same with the experiments reported in (Su et al., 2024) for fair comparison and all the retrievers use the original queries in BRIGHT to retrieve documents. Since Seed1.5-Embedding is not public available when this work is done, we directly use the experiment results reported on their model card.

Query Reasoner with BM25 We include two types of baselines using state-of-the-art large language models: 1) Non-reasoning models including GPT-4o, doubao-1.5-pro [5], DeepSeek-V3 (DeepSeek-AI et al., 2025); 2) Reasoning models including DeepSeek R1 (Guo et al., 2025), o1-mini[6], o1-preview[7], DeepSeek-R1-Distill-Qwen-7B[8], DeepSeek-R1-Distill-Qwen-32B[9] and QwQ-32B (Qwen et al., 2025). All the models use the prompt in Appendix A for reaoning. For each baseline, we only remain the prediction result of after reasoning and use BM25 for further retrieval.

Query Reasoner with Reasoning-Intensive Retrievers (ReasonIR) ReasonIR (Shao et al., 2025) is the most recently acknowledged retriever specifically trained for reasoning-intensive retrieval tasks. We further combine RQR with ReasonIR for comparsion to explore further improvements with the specialized reasoner and retriever in this task.

[4]https://huggingface.co/ByteDance-Seed/Seed1.5-Embedding

[5]https://seed.bytedance.com/en/special/doubao_1_5_pro

[6]https://openai.com/index/openai-o1-mini-advancing-cost-efficient-reasoning/

[7]https://openai.com/index/introducing-openai-o1-preview/

[8]https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-7B

[9]https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-32B

4.1.3 Implementation Details

With the initial checkpoint of Qwen2.5-7B-Instruct[10] and Qwen2.5-1.5B-Instruct[11], RQR 7B and 1.5B are both trained with TRL[12] on a single node with 4 NVIDIA A800-80G GPUs. Following the instructions of Open-R1 (Face, 2025), we use 1 GPU for vLLM (Kwon et al., 2023) serving and the rest 3 GPUs for model training. DeepSpeed (Rasley et al., 2020) ZeRO-3 and Gradient Checkpoint are applied to reduce the cost of VRAM. It takes about 16 hours for 1.5B model training and about 48 hours for 7B model training. We set the learning rate $1e-6$, the batch size per device 16, and the KL coefficient 0.008. For each input prompt, 16 samples are generated to estimate the advantage in GRPO. Since we use bge-base-en-v1.5[13] embedding model to compute relevance, the maximum completion length is set to 500 to avoid exceeding the input length limitation of the embedding model. Experiments on all the above-mentioned baselines are conducted without reranking.

4.2 Main results

Table 1 shows that our 7B model outperforms all query reasoning baselines of non-reasoning LLMs, including GPT-4o and DeepSeek V3, performing comparable to the large reasoning models, *e.g.*, o1-mini, QwQ-32B and DeepSeek R1. Our 7B model strikes a favorable balance between inference efficiency and reasoning performance, offering a compelling trade-off for query reasoning tasks. Besides, our 1.5B model also achieves performance comparable to that of large-scale language models, making it an effective solution for resource-constrained scenarios.

To quantitatively assess the efficiency of different models, we report both their *Performance* and *Cost* in Table 2. Here, **Performance** is defined as the nDCG@10 score achieved by each model on the BRIGHT benchmark (Su et al., 2024) with BM25 retriever. Meanwhile, **Cost** represents the price of each model (USD per 1M output tokens) when accessed via the OpenRouter platform[14], indicating the actual monetary expense required to obtain outputs from the model[15]. Based on the calculated efficiency (*Eff* = *Performance* / *Cost*), RQR-1.5B and RQR-7B achieve the highest cost-effectiveness among all evaluated models, with efficiency scores of **2460.0** and **279.0**, respectively. This highlights the strong cost-performance advantage of the our method.

Model	Performance	Cost	Efficiency
GPT-4o	26.5	10.0	2.7
DeepSeek V3	27.8	0.9	31.6
DeepSeek R1	29.6	2.2	13.6
QwQ-32B	28.9	0.2	144.5
o1-preview	30.2	60.0	0.5
o1-mini	28.3	4.4	6.4
RQR-7B	27.9	0.1	279.0
RQR-1.5B	24.6	0.01	**2460.0**

Table 2: Model performance (Perf.), cost, and efficiency (Eff. = Perf. / Cost).

Compared to the retrievers specifically trained for reasoning, both the 7B and 1.5B models can both outperform ReasonIR with original query, and our 7B model can outperform Seed 1.5-Embedding. Since ReasonIR is an embedding model based on the backbone of LLaMA3.1-8B, the computational cost of pre-encoding documents can be prohibitive when the corpus is large. In contrast, RQR can work with BM25 retrievers, incurring significantly lower pre-processing costs than LLM-based embedding models.

As ReasonIR can work with reasoned queries to achieve better performance, we apply the reasoned queries generated by our method for further exploring the effect of combining reasoned queries with reasoning-intensive retrievers. We confirmed that our method can achieve further improvements based on ReasonIR. Our 1.5B model can outperform LLama3.1-8B-Instruct which is more than 5 times larger in parameters, and our 7B model can outperform GPT-4. Comparing with the reasoned queries of GPT-4 and our 7B model , the performance improvement based on ReasonIR (29.9->31.9) is higher than the improvement based on BM25 (26.5->27.9). These results incidate that our method is flexible and can work with different retrievers, the improvement of retriever will further expand the advantage of our method.

4.3 Ablation studies

4.3.1 Effect of Data Size and Quality

DeepSeek R1 performs better than QwQ-32B while the performance of V1-R1 and V1-QwQ is close

[10] https://huggingface.co/Qwen/Qwen2.5-7B-Instruct
[11] https://huggingface.co/Qwen/Qwen2.5-1.5B-Instruct
[12] https://github.com/huggingface/trl
[13] https://huggingface.co/BAAI/bge-base-en-v1.5
[14] https://openrouter.ai
[15] We use the price of Qwen2.5-7B-Instruct as the price of RQR-7B, and we define the price of RQR-1.5B as 0.01 since the price of Qwen2.5-1.5B is free.

Dataset	Bio.	Earth.	Econ.	Psy.	Rob.	SO	SL	LC	Pony	AoPS	TQ	TT	Avg
V1-QwQ	42.3	43.6	19.1	**31.9**	18.6	**23.7**	22.8	21.5	17.7	4.0	17.2	10.1	22.7
V1-R1	**48.0**	46.5	19.9	31.4	15.2	23.6	22.3	21.2	18.0	**5.3**	18.8	11.3	23.5
V2	46.0	**47.1**	**21.1**	31.2	**19.8**	21.7	**24.3**	**22.5**	**21.8**	4.3	**19.7**	**15.9**	**24.6**

Table 3: Results on different training data for RQR-1.5B with BM25 retriever.

on most BRIGHT subtasks. V1-R1 exhibits a notable advantage only in the Biology and Earth Science subtasks. We hypothesize that this may be attributed to the fact that these two subtasks are more knowledge-intensive compared to others, thereby granting the larger-parameter DeepSeek R1 model with 671B parameters a more pronounced advantage over QwQ-32B. The V2 dataset with more samples leads to the best performance. Instead of using large reasoning models to generate answers for distillation, a better approach may be to use the answers selected by the users in the StackExchange datasets. It can be easily scaled since generating answers with large reasoning models on large-scale question set is too expensive.

4.3.2 Effect of Reinforcement Learning with Semi-Rule-Based Rewards

We further explore the effect of our proposed approaches with semi-rule-based reward functions compared to traditional supervised fine-tuning (SFT). Following the same experimental settings in Section 4.3.1, we use Qwen2.5-1.5B-Instruct, with BM25 as retrievers. With the dataset of V1-QwQ and V2, we separately trained the model with SFT and RL. Results are shown in Table 4, where "RL" is for our proposed reinforced learning approaches and "SFT" is for supervised fine-tuning. Both RL and SFT are in full parameters.

These results indicate that when using the training data generated by large reasoning models, the performance of RL is slightly higher than SFT. - While using the user-selected answer data for training, the performance of SFT experienced a significant decline. This is likely because the user-selected answers written by actual users may exhibit substantial quality deficiencies (*e.g.*, higher perplexity) compared to data synthesized by large reasoning models. In addition, we did not apply fine-grained data cleaning for the answer. As a result, the answers of the questions may include URL links of pictures which do not include available information. Using such data for supervised fine-tuning may lead to catastrophic forgetting in the model. In contrast, our proposed reinforcement learning approach with semi-rule-based reward functions does not strictly require the model to fit the answers per token exactly. Since the relevance score in reward function is based on the embedding similarity of generated answers and selected answers, the noisy signals in selected answers may not explicitly affect the similarity scores. As a result, our proposed approach demonstrates stronger generalization capabilities and greater tolerance for noisy data.

Training Data	Training Method	Avg
V1-QwQ	SFT	22.4
V1-QwQ	RL	22.7
V2	SFT	12.8
V2	RL	24.6

Table 4: Results on different training data and methods.

4.3.3 Effect of Relevance Model in Reward Functions

As we mentioned in Section 3.2, the relevance model is playing an important role in our proposed semi-rule-based reward functions. We further explore the effect of different relevance models in our proposed reward functions. Besides the dense embedding model of bge-base-en-v1.5, we also implement the relevance function via the sparse model of bge-m3 (Chen et al., 2024a). As the bge-m3 model can accept a longer input length, we also explore the effect of extending the maximum completion length to 1000. With Qwen2.5-1.5B-Instruct as base model and BM25 as retriever, we train the model on different reward functions and completion length settings on the training data of V1-R1. Results are shown in Table 5.

Since all experiments are conducted with the sparse retriever of BM25, we initially expected bge-m3, as a sparse relevance model, to offer performance improvements. However, bge-m3 actually underperforms compared to the dense embedding model bge-base-en-v1.5, which has fewer param-

Model	Type	Length	Avg
bge-base-en-v1.5	Dense	500	23.5
bge-m3	Sparse	500	23.1
bge-m3	Sparse	1000	22.9

Table 5: Results on different relevance model types and competition length settings.

eters (110M vs 550M)[16]. This result suggests that for our proposed semi-rule-based reward function, overly fine-grained relevance matching signals may harm the model's generalization ability. It is worth noting that our training data is based on V1-R1 rather than V2, these results are unlikely to be primarily attributed to data noise since the answers are generated by DeepSeek R1. Furthermore, we observed that even increasing the output length did not improve performance, indicating that excessively long outputs might dilute the effective relevance signals, thus providing no benefit to final retrieval performance.

4.3.4 Effect of Explicit Thinking

Content	Explicit Thinking	Avg
Answer	No	22.7
Thinking+Answer	Yes	21.4
Answer	Yes	20.9

Table 6: Results on explicit thinking process.

Inspired by DeepSeek R1 (Guo et al., 2025) and some recent works (Weller et al., 2025; Xie et al., 2025), we further investigate the effect of explicit thinking process. When the explicit thinking process is applied, the model will first think about the reasoning process explicitly and then provide the actual answer. The reasoning process and answer are enclosed with "<think></think>" and "<answer></answer>" tags. With the dataset of V1-QwQ, we train the model on Qwen2.5-1.5B-Instruct, and evaluate the query reasoners with BM25 retriever. Since the thinking process requires external output tokens, the max completion length is set to 1000 when the explicit thinking process is applied. Details about the prompt and reward settings are listed in Appendix C. Results are shown in Table 6. In the table, "Explicitly Thinking" denotes if the explicitly thinking process is applied for model training, and "Content" denotes if the output query contains the thinking process within the "<think></think>" tags. "Thinking+Answer" means that the contents within the "<think></think>" and "<answer></answer>" tags are concatenated as the reasoned queries, and "Answer" means that only the answer content is returned.

Experimental results indicate that applying explicit thinking process does not improve performance on query reasoning tasks. Previous studies (Weller et al., 2025) have shown that explicitly generating the reasoning process within the "<think></think>" tags can be beneficial for certain reasoning-intensive tasks, possibly because these tasks require the model to produce answers in specific output formats. For example, in ranking tasks, the model receives a query and a document as input and must output a binary relevance judgment (true or false). In such cases, applying explicit thinking process can help the model fully leverage its reasoning capabilities through chain-of-thought prompting, thereby enhancing inference performance. However, in the case of query reasoning tasks, the generated reasoned query inherently encapsulates the reasoning process and is not constrained by output format requirements. As a result, explicitly generating the reasoning process does not lead to further performance gains.

5 Conclusion

In this work, we present RQR, a family of compact and efficient language models tailored for query reasoning and rewriting in reasoning-intensive retrieval. By leveraging the learning algorithm of GRPO with a novel semi-rule-based reward function, our approach enables effective and robust reinforcement learning without relying on expensive human-annotated datasets and retrieval sources. Our proposed models demonstrate strong performance on the BRIGHT benchmark, rivaling or even surpassing large-scale commercial LLMs, while significantly reducing inference cost and latency. Furthermore, RQR models exhibit strong compatibility with both traditional and reasoning-intensive retrievers, making them highly versatile for real-world deployment. Our findings highlight a promising direction toward building lightweight, affordable, and high-performing reasoning components for retrieval-augmented generation pipelines and the latest deep research products.

[16]bge-m3 is based on XLM-RoBERTa-Large

Limitations

Our work still has several limitations that we plan to address in future works:

- Besides reasoning-intensive retrieval, due to the limitation of time and computational cost, we omit the effect of query reasoning in other reasoning-intentsive RAG tasks including MMLU (Hendrycks et al., 2021) and GPQA (Rein et al., 2024).
- We directly used the publicly available StackExchange dataset to build our training data, and we did not wash the answers carefully. Although our proposed approach may not be easily affected by the noisy training data, it may still be beneficial to use a high-quality training set.
- By the time this work is done, the latest Qwen3[17] model family is released. Replacing the initial checkpoints to Qwen3-1.7B, 8B and 14B may lead to further improvements.

Ethics Statement

References

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2018. Ms marco: A human generated machine reading comprehension dataset. *Preprint*, arXiv:1611.09268.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024a. BGE M3-Embedding: Multi-Lingual, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation. *arXiv preprint*. ArXiv:2402.03216 [cs].

Lingjiao Chen, Jared Quincy Davis, Boris Hanin, Peter Bailis, Ion Stoica, Matei Zaharia, and James Zou. 2024b. Are more llm calls all you need? towards scaling laws of compound inference systems. *Preprint*, arXiv:2403.02419.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2025. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

[17] https://qwenlm.github.io/blog/qwen3/

Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Hugging Face. 2025. Open r1: A fully open reproduction of deepseek-r1.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Pengcheng Jiang, Jiacheng Lin, Lang Cao, Runchu Tian, SeongKu Kang, Zifeng Wang, Jimeng Sun, and Jiawei Han. 2025. Deepretrieval: Hacking real search engines and retrievers with large language models via reinforcement learning. *Preprint*, arXiv:2503.00223.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. *Preprint*, arXiv:2309.06180.

Nathan Lambert, Lewis Tunstall, Nazneen Rajani, and Tristan Thrush. 2023. Huggingface h4 stack exchange preference dataset.

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. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 364.

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, SIGIR '24, pages 2421–2425, New York, NY, USA. Association for Computing Machinery.

Tong Niu, Shafiq Joty, Ye Liu, Caiming Xiong, Yingbo Zhou, and Semih Yavuz. 2024. Judgerank: Leveraging large language models for reasoning-intensive reranking. *Preprint*, arXiv:2411.00142.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, and 401 others. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 3505–3506, New York, NY, USA. Association for Computing Machinery.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2024. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Rulin Shao, Rui Qiao, Varsha Kishore, Niklas Muennighoff, Xi Victoria Lin, Daniela Rus, Bryan Kian Hsiang Low, Sewon Min, Wen tau Yih, Pang Wei Koh, and Luke Zettlemoyer. 2025. Reasonir: Training retrievers for reasoning tasks. *Preprint*, arXiv:2504.20595.

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

Hongjin Su, Howard Yen, Mengzhou Xia, Weijia Shi, Niklas Muennighoff, Han-yu Wang, Haisu Liu, Quan Shi, Zachary S Siegel, Michael Tang, and 1 others. 2024. Bright: A realistic and challenging benchmark for reasoning-intensive retrieval. *arXiv preprint arXiv:2407.12883*.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2023. Improving text embeddings with large language models. *arXiv preprint arXiv:2401.00368*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

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

Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. 2024. Inference scaling laws: An empirical analysis of compute-optimal inference for llm problem-solving. In *The 4th Workshop on Mathematical Reasoning and AI at NeurIPS'24*.

Tian Xie, Zitian Gao, Qingnan Ren, Haoming Luo, Yuqian Hong, Bryan Dai, Joey Zhou, Kai Qiu, Zhirong Wu, and Chong Luo. 2025. Logic-rl: Unleashing llm reasoning with rule-based reinforcement learning. *Preprint*, arXiv:2502.14768.

Siyun Zhao, Yuqing Yang, Zilong Wang, Zhiyuan He, Luna K Qiu, and Lili Qiu. 2024. Retrieval augmented generation (rag) and beyond: A comprehensive survey on how to make your llms use external data more wisely. *arXiv preprint arXiv:2409.14924*.

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

A Prompt Templates

Figure 2 shows the prompt template for the instructions of chain-of-thought query reasoning. The reasoner model takes the instructions and original query as input, and return a "pseudo-answer" with thoughts including as much relevant information as possible. The "pseudo-answer" can be used as the reasoned query, and the retriever can benefit from the external information provided by the reasoned query.

Instruction Templates for Query-Reasoning

```
Instructions:
1. Identify the essential problem.
2. Think step by step to reason and describe what information could be relevant
and helpful to address the questions in detail.
3. Draft an answer with as many thoughts as you have
Query: {query}
```

Figure 2: The prompt template for the instructions of Chain-of-Thought query reasoning.

B Training Data

Details about the construction of training data are described as follows:

- Version 1: sampling at most 1200 questions for each selected category to generate answers with large reasoning models. The selected categories include: 'biology', 'chemistry', 'codereview', 'cs', 'earthscience', 'economics', 'math', 'physics', 'robotics'. The Version 1 dataset includes around 10k sampled questions. In this paper, the corresponding datasets are denoted as **V1-R1** and **V1-QwQ**, indicating that the answers are generated by DeepSeek R1 or QwQ-32B.

- Version 2: sampling at most 1500 questions for each selected category with selected answers. Those questions can also be used to generate answers via large reasoning models. The categories include: 'ai', 'biology', 'chemistry','codereview', 'cs', 'earthscience', 'economics', 'computergraphics', 'math', 'mathoverflow', 'philosophy', 'physics', 'robotics', 'stackoverflow', 'sustainability', 'softwareengineering', 'bioinformatics'. The Version 2 dataset includes around 30k sampled questions, nearly three times as many as Version 1, making answer generation with large reasoning models unaffordable since the inference time is too long. In this paper, the dataset is denoted as **V2**.

C System Prompt and Reward for Explicit Thinking

Inspired by previous works (Xie et al., 2025; Weller et al., 2025), we use the following system prompt to instruct the model to output the thinking process explicitly in the format of "<think>thinking process</think><answer>the answer</answer>".

System Prompt

You are a helpful assistant. The assistant first thinks about the reasoning process in the mind and then provides the user with the answer. The reasoning process and answer are enclosed within `<think> </think>` and `<answer> </answer>` tags, respectively, i.e., `<think>` reasoning process here `</think> <answer>` answer here `</answer>`.

When the explicit thinking process is applied, we also design a format reward to force the model returning an output in the correct format. Our format checking strategy is identical to (Xie et al., 2025). If

the model's output fails the format checking, the reward function will immediately return a score of -1, and the subsequent computation of the query reasoning reward will be skipped.

D License

In this section we list the artifacts we used and the corresponding URL and licenses:

Name	Type	URL	License
StackExchange-Preferences	Dataset	https://huggingface.co/datasets/HuggingFaceH4/stack-exchange-preferences	cc-by-sa-4.0
BRIGHT Benchmark	Dataset	https://huggingface.co/datasets/xlangai/BRIGHT	cc-by-4.0
Qwen2.5-1.5B-Instruct	Model	https://huggingface.co/Qwen/Qwen2.5-1.5B-Instruct	apache-2.0
Qwen2.5-7B-Instruct	Model	https://huggingface.co/Qwen/Qwen2.5-7B-Instruct	apache-2.0
bge-base-en-v1.5	Model	https://huggingface.co/BAAI/bge-base-en-v1.5	mit
bge-m3	Model	https://huggingface.co/BAAI/bge-m3	mit
QwQ-32B	Model	https://huggingface.co/Qwen/QwQ-32B	mit
DeepSeek R1	Model	https://huggingface.co/deepseek-ai/DeepSeek-R1	mit
DeepSeek V3	Model	https://huggingface.co/deepseek-ai/DeepSeek-V3-0324	mit
ReasonIR	Model	https://huggingface.co/reasonir/ReasonIR-8B	cc-by-nc-4.0

Table 7: List of datasets and models used, along with their URLs and licenses.