YOUR DENSE RETRIEVER IS SECRETLY AN EXPEDITIOUS REASONER

Anonymous authors
Paper under double-blind review

ABSTRACT

Dense retrievers enhance retrieval by encoding queries and documents into continuous vectors, but they often struggle with reasoning-intensive queries. Although Large Language Models (LLMs) can reformulate queries to capture complex reasoning, applying them universally incurs significant computational cost. In this work, we propose Adaptive Query Reasoning (AdaQR), a hybrid query rewriting framework. Within this framework, a Reasoner Router dynamically directs each query to either fast dense reasoning or deep LLM reasoning. The dense reasoning is achieved by the Dense Reasoner, which performs LLM-style reasoning directly in the embedding space, enabling a controllable trade-off between efficiency and accuracy. Experiments on large-scale retrieval benchmarks BRIGHT show that AdaQR reduces reasoning cost by 28% while preserving—or even improving—retrieval performance by 7%[1].

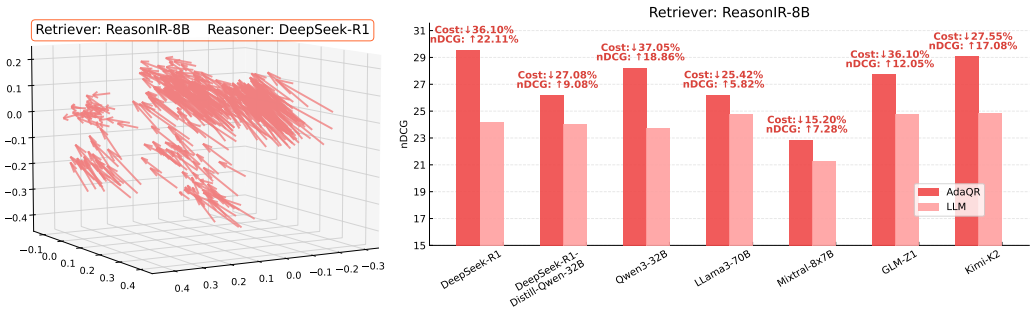

Figure 1: *Left*: PCA-reduced visualization of embedding transformation, where each arrow denotes the shift from the original query embedding to its reasoned counterpart, with most following systematic and structured trajectories. *Right*: AdaQR on the ReasonIR-8B dense retrievers yields substantial gains in retrieval performance and query rewriting efficiency on BRIGHT benchmark.

1 INTRODUCTION

Information Retrieval (IR) is a fundamental technology that bridges user queries and relevant documents across vast corpora. It plays a pivotal role in search engines, question answering system, etc (Bai et al., 2022; Muennighoff et al., 2023; Jin et al., 2025). Traditional approaches primarily rely on keyword matching (Robertson & Zaragoza, 2009) to evaluate relevance. While effective, these methods often struggle with capturing deeper semantics and contextual nuances (Chen et al., 2024a). Dense retrieval methods address these limitations by encoding queries and documents into continuous vector representations and performing similarity search in the embedding space. These approaches enable the retrieval system to handle contextually complex queries, and significantly improve recall performance (Zhao et al., 2024; Lin et al., 2025b; Zhang et al., 2025c).

However, for many reasoning-intensive real-world queries, the conventional embeddings produced by dense retrievers often fail to capture the relevance between a query and the retrieved documents—relevance that only becomes evident after further reasoning (Su et al., 2025; Chen et al.,

[1]Our code will be released to facilitate further research.

2025a). Recently, Large Language Models (LLMs), thanks to their increasingly powerful reasoning capabilities, have been employed to reformulate original queries, substantially enhancing performance in reasoning-intensive retrieval scenarios (Kostric & Balog, 2024; Dharwada et al., 2025). Nevertheless, applying LLM-based query rewriting to all queries in online or large-scale retrieval systems incurs substantial computational and latency costs, thereby becoming a bottleneck (Nguyen et al., 2025; Qin et al., 2025).

The primary cost of LLM reasoning arises from its auto-regressive and often lengthy generation process (Lin et al., 2025a). Consequently, a natural direction for optimization is to avoid performing this explicit reasoning step altogether. This raises an interesting question: *Is it possible for the reasoning process to be carried out implicitly?* Fortunately, an affirmative answer has been given to this question by recent latent reasoning studies, in which a model performs reasoning implicitly within its internal representations (Li et al., 2025; Chen et al., 2025b). However, these methods still rely on LLMs as the reasoner and can't meet the efficiency needs in high-throughput retrieval scenarios (Shen et al., 2025). A more efficient solution is to achieve this latent reasoning process directly by the dense retriever, which is feasible as we observe that: *the embeddings of some queries before and after LLM reasoning exhibit systematic, structured transformation* (see Figure 1, left). To this end, we propose the Dense Reasoner (DR), which learns to perform LLM-style query reasoning directly in the embedding space at negligible cost, resulting in extremely fast retrieval.

Nevertheless, certain queries may not be adequately addressed by the structured transformation of dense reasoning. In such cases, retaining LLM-based reasoning is essential to ensure robust overall retrieval performance, thus a routing mechanism is needed to ensure appropriate reasoning process (Bai et al., 2024; Zhang et al., 2025a). To this end, we introduce the Reasoner Router (RR). Given a new query, the Reasoner Router determines whether it can be reliably handled by the Dense Reasoner, i.e., whether the learned structured transformation can stably approximate the semantic effect of LLM reasoning, and otherwise resorts to LLM for deep reasoning. This mechanism enables a controllable trade-off between efficiency and accuracy. Built on the components described above, we propose the Adaptive Query Reasoning (AdaQR) framework. AdaQR is a hybrid reasoning pipeline consisting of three components: an LLM Reasoner, a Dense Reasoner, and a Reasoner Router. The Reasoner Router flexibly directs each user query to either the Dense Reasoner or the LLM Reasoner, preserving—and in some cases improving—the retrieval quality achieved by full LLM rewrites, while substantially reducing the reasoning (see Figure 1, right).

In summary, our contributions are as follows:

- We propose a novel **AdaQR** framework that enables a hybrid fast-and-deep query reasoning strategy, preserving retrieval performance while significantly reducing rewriting cost.
- Within AdaQR, we introduce the **Dense Reasoner**, which imitates LLM reasoning in the embedding space, achieving extremely fast query rewriting.
- We further propose the **Reasoner Router** to schedule query reasoning within AdaQR. It appropriately directs each query to fast dense reasoning or deep LLM reasoning.
- Experiments on BRIGHT (Su et al., 2025) benchmark demonstrate that AdaQR generalizes well across diverse settings. Compared to serving a full LLM, our approach achieves an average 7% improvement in retrieval performance while reducing the rewriting cost by an average of 28%.

2 Methodology

2.1 Problem Statement

Given a query q, a fixed corpus $\mathcal{D} = \{d_1, d_2, \cdots, d_{|\mathcal{D}|}\}$ with $|\mathcal{D}|$ documents and a set of relevant documents $\mathcal{G}_q \subset \mathcal{D}$ of query q, a retrieval system $\mathcal{R}$ generates the relevance score s_i for each document d_i and ranks the top-k documents based on their scores, expecting that the relevant document in $\mathcal{G}_q$ appears higher in the rank list:

$$\mathcal{R}(q, \mathcal{D}) = \{(d_{i_1}, s_{i_1}), (d_{i_2}, s_{i_2}), \cdots, (d_{i_k}, s_{i_k})\}, s_{i_1} > s_{i_2} > \cdots > s_{i_k} \tag{1}$$

In modern dense retrieval, the crucial component is a powerful embedding model $\mathcal{E} : \text{text} \to \mathbb{R}^n$, mapping queries and documents into a n-dimensional shared continuous vector space. Let $e_q =$

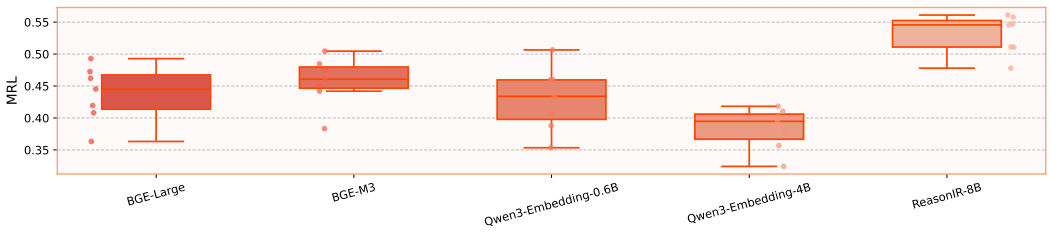

Figure 2: Distributions of MRL scores for LLM Reasoners, with each point representing the score computed for a given LLM Reasoner and dense retriever pair.

$\mathcal{E}(q)$ and $e_d = \mathcal{E}(d)$ denote the representations corresponding to query q and document d. A dense retrieval system is then performed by computing the relevance score between e_q and e_d using a similarity function (e.g., cosine similarity, euclidean distance).

When meeting challenging and reasoning-intensive queries, one typically needs to rewrite them to reason their intrinsic solutions by a query reasoner, typically a powerful LLM $\mathcal{M}_{\text{LLM}}$. It transforms the original query into a reasoned query in text form $q^{\text{LLM}} = \mathcal{M}_{\text{LLM}}(q)$. The reasoned queries typically convey richer and more explicit semantic content than the original query, and therefore replace the original query in the following retrieval steps to achieve better retrieval results.

2.2 Pilot Study

AdaQR is based on a simple, empirically informed hypothesis: for part of queries, the semantic transformation induced by LLM reasoning manifest as systematic, structured transformation in the embedding space rather than random, disordered variations.

To further verify this hypothesis, we conduct a simple experiment on BRIGHT (Su et al., 2025). Specifically, we generate the reasoned queries by employing 7 different LLM Reasoners. We then encode the original queries and all their reasoned counterparts with 5 dense retrievers. To quantify the directional coherence between the embeddings of original and reasoned queries, we compute the Mean Resultant Length (MRL) (Kutil, 2012). Concretely, for all embedding pairs $\mathcal{P} = \{(e_{q_1}, e_{q_1^{\text{LLM}}}), (e_{q_2}, e_{q_2^{\text{LLM}}}), \cdots, (e_{q_{|\mathcal{P}|}}, e_{q_{|\mathcal{P}|}^{\text{LLM}}})\}$, MRL is defined as:

$$\text{MRL}(\mathcal{P}) = \frac{1}{|\mathcal{P}|} \cdot || \sum_i \frac{e_{q_i^{\text{LLM}}} - e_{q_i}}{||e_{q_i^{\text{LLM}}} - e_{q_i}||} ||, \tag{2}$$

where $||\cdot||$ represent the L2 norm. The MRL results are summarized in Figure 2. Apparently, across all LLM Reasoners and dense retrievers, the MRL remains relatively high with an average value 0.45, indicating a substantial degree of agreement in the transformation induced by LLM-driven reasoning. Notably, different Dense Retriever exhibits different MRL value, with higher MRL value easier to learn embedding transformation, which is further corroborated in Section 3.2.

2.3 Adaptive Query Reasoning

Based on the above hypothesis, we propose **Ada**ptive **Q**uery **R**easoning (**AdaQR**) framework, a hybrid pipeline designed to retain the retrieval performance with LLM reasoning while dramatically reducing the reasoning cost. As shown in Figure 3, our approach presents a comprehensive framework for query reasoning, encompassing three complementary components: a LLM Reasoner, a Dense Reasoner, and an Reasoner Router.

Specifically, the LLM Reasoner is for typical query rewriting based on the LLM reasoning ability, effective but at a high cost. The Dense Reasoner is an embedding transformation optimized to approximate the semantic transformation of the LLM Reasoner's rewrites with a negligible cost, serving as an efficient alternative for LLM reasoning. Then, the reasoner router acts as a decision mechanism to route each query either to the low-cost dense reasoning or to the LLM reasoning, depending on the predictability of its LLM reasoned counterpart.

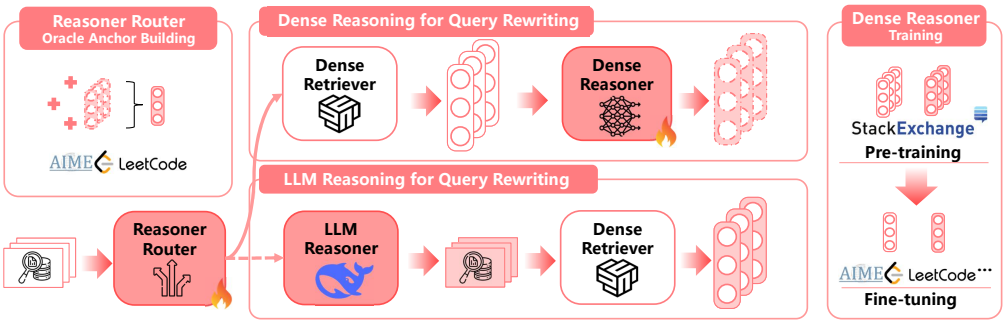

Figure 3: Overview of AdaQR: the Reasoner Router directs each input query through the oracle anchor to either the LLM Reasoner or the Dense Reasoner for reasoning-based query rewriting. The Dense Reasoner is constructed via adjacent pre-training and fine-tuning, enabling it to efficiently imitate the reasoning behavior of the LLM in the embedding space.

2.3.1 Dense Reasoner

The Dense Reasoner expects to reproduce the semantic transformation induced by LLM reasoning at negligible inference cost, so that its resultant embeddings can be used directly for the following retrieval steps. We achieve this by applying a compact parametric mapper and employing a two-stage training strategy.

For original query q and reasoned query q^{LLM} generated by LLM, along with their d-dimensional embedding vector e_q and $e_{q^{\text{LLM}}}$, the Dense Reasoner is formalized as a parameterized mapping $\mathcal{M}_{\text{DR}}(e;\theta) : \mathbb{R}^d \to \mathbb{R}^d$ (where θ are the trainable parameters) that generates a reasoned embedding $\hat{e}_q = M_{\text{DR}}(e_q;\theta)$ intended to approximate $e_{q^{\text{LLM}}}$. The training objective is to find parameters θ minimizing the mean squared error (MSE) between predicted and target embeddings:

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{M} \sum_{i=1}^{M} \left\| \mathcal{M}_{\text{DR}}(e_{v_i};\theta) - e_{v_i^{\text{LLM}}} \right\|_2^2 \tag{3}$$

To balance generalization and domain adaptation, we adopt a **two-stage training strategy**. For the first stage, we pre-train the Dense Reasoner on large-scale embedding pairs of original and reasoned queries to learn general reasoning transformation patterns. For the second stage, to adapt to in-domain distributions while avoiding catastrophic forgetting, we fine-tune the Dense Reasoner on the downstream datasets using a reduced learning rate and epoch.

Throughout a two-stage training strategy, Dense Reasoner learns the dominant embedding transformation induced by high-quality LLM Reasoners. Benefit from the compact mapper structure, Dense Reasoner generates reasoning embeddings $\hat{v}_q$ with low-cost forward pass, avoiding LLM invocations and dense retrievers' cost while largely preserving ranking quality for queries.

2.3.2 Reasoner Router

The Dense Reasoner provides a low-cost approximation of LLM-based query reasoning, while LLM Reasoner provides higher-quality but substantially more expensive textual rewrites. The Reasoner Router is a lightweight routing mechanism that selects the most appropriate reasoning pathway for the rewriting of each query. Concretely, Reasoner Router makes a judgment whether to apply the low-cost reasoning produced by Dense Reasoner or to roll back to the LLM reasoning, enabling a controllable trade-off between total cost and retrieval effectiveness.

For each query, we use an oracle anchor to measure the predictability of its LLM reasoning result. Concretely, we consider training queries whose reasoning performance by Dense Reasoner is comparable to or better than that of LLM reasoner as $\mathcal{S}$. We build the oracle anchor for this set, which captures the shared semantic and intent signals of queries that exhibit predictable, learnable embedding-space reasoning:

$$p = \frac{1}{|S|} \sum_{q \in S} e_q \tag{4}$$

At reasoning time, the Dense Reasoner compares a query similarity $\mathrm{sim}(e_q, p)$ to a threshold τ, representing how query is likely to benefit from the low-cost Dense Reasoner. The final embedding $\tilde{e}_q$ for retrieval is defined as:

$$\tilde{e}_q = \begin{cases} \mathcal{M}_{\mathrm{ERR}}(e_q; \hat{\theta}), & \text{if } \mathrm{sim}(e_q, p) \geq \tau, \\ e_{q^{\mathrm{LLM}}}, & \text{otherwise.} \end{cases} \tag{5}$$

Benefit from this flexible selection mechanism, the Reasoner Router directs each query to the most appropriate reasoning path. This component not only avoids the suboptimal retrieval caused by purely applying Dense Reasoning to unstructured queries, but also balances computational cost and retrieval quality through the adaptive threshold τ to meet resource constraints.

3 Experiments

3.1 Experiment Setting

Evaluation Dataset We employ BRIGHT (Su et al., 2025), a reasoning-intensive retrieval benchmark. It contains 1,385 real-world queries from a variety of domains (StackExchange, LeetCode, and math competitions, etc.), typically requiring deliberate semantic reasoning to match. These properties transformation BRIGHT a challenging and realistic choice for rewritten-query retrieval.

Metrics Following the original BRIGHT setup, we adopt the average nDCG@10 across the 12 datasets in BRIGHT as our evaluation metric. Formally, for a single query, the normalized discounted cumulative gain (nDCG) at rank k is defined as:

$$\mathrm{nDCG@}k = \frac{\mathrm{DCG@}k}{\mathrm{IDCG@}k} \quad \text{with} \quad \mathrm{DCG@}k = \sum_{i=1}^{k} \frac{2^{\mathrm{rel}_i} - 1}{\log_2(i+1)} \tag{6}$$

where rel_i is the relevance of the document at rank i, and $\mathrm{IDCG@}k$ is the maximum possible DCG@k for the query.

Pre-training of Dense Reasoner We construct an external corpus from StackExchange [2], a popular community-driven platform. The corpus contains 10k reasoning-intensive questions (see details in Appendix A.2). We perform careful curation and filtering to ensure high-quality data and no overlap with the BRIGHT dataset. This external corpus is intended to teach the Dense Reasoner general, cross-domain reasoning transformation patterns before in-domain adaptation.

Fine-tuning of Dense Reasoner & Oracle Anchor Building We partition BRIGHT into an in-domain training portion and a held-out test portion, following common practice Chen et al. (2024b); Zhuang et al. (2025); Zhang et al. (2025d): 70% for fine-tune the Dense Reasoner and building oracle anchor in Reasoner Router, and 30% for evaluation.

LLM Reasoners We employ 17 widely-used LLMs for query reasoning as follows:

- **DeepSeek**: DeepSeek-R1 (DeepSeek-AI et al., 2025), DeepSeek-V3 (DeepSeek-AI et al., 2024), DeepSeek-R1-Distill-Qwen-32B / 14B / 7B (DeepSeek-AI et al., 2025), DeepSeek-R1-Distill-Llama-70B / 8B (DeepSeek-AI et al., 2025).
- **Qwen**: Qwen3-32B / 14B / 7B / 4B (Yang et al., 2025).
- **Meta**: LLama3-70B (Dubey et al., 2024).
- **Mistral**: Mixtral-8x7B (Jiang et al., 2024), Mistral-7B (Jiang et al., 2023).
- **ZAI**: GLM-Z1 (Zeng et al., 2024), GLM-4 (Zeng et al., 2024).
- **MoonShot**: Kimi-K2 (Bai et al., 2025).

Dense Retrievers We employ 5 leading dense retrieval models for encoding query and retrieval, including BGE-Large (Xiao et al., 2023), BGE-M3 (Chen et al., 2024a), Qwen3-Embedding-0.6B, Qwen3-Embedding-4B (Zhang et al., 2025c) and ReasonIR-8B (Shao et al., 2025), a SOTA retriever specialized for reasoning-intensive retrieval.

[2] https://huggingface.co/datasets/HuggingFaceH4/stack-exchange-preferences

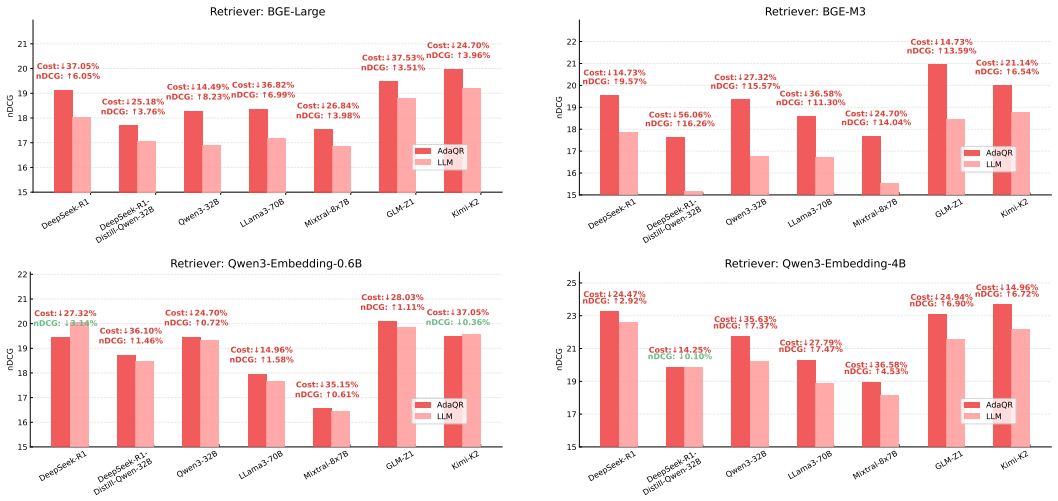

Figure 4: Retrieval performance improvement and reasoning cost reduction on the BRIGHT held-out test set achieved by AdaQR compared to LLM-based reasoning.

Implementation Details We implement the Dense Reasoner with a lightweight two-layer MLP. The hidden size is set equal to the input embedding dimension, and each hidden layer is followed by a tanh activation. We pre-train it for 50 epochs with a learning rate of 5×10^{-4} and fine-tune for 3 epochs with a smaller learning rate 1×10^{-5} to adapt to in-domain distributions. Since different dense retrievers exhibit varying representational capacities and geometric properties, the choice of the threshold parameter τ in reasoner router is different. In practice, we determine τ empirically (e.g. 0.75 for BGE-Large, 0.7 for BGE-M3 and ReasonIR-8B, 0.6 for Qwen3-Embedding-0.6B and Qwen3-Embedding-4B).

3.2 Main Result

We present the comparisons of AdaQR and LLM reasoning in terms of performance and cost in Figure 1 and Figure 4. Our analysis reveals several significant findings:

AdaQR achieves better performance at lower cost. Compared to LLM reasoning, AdaQR achieved higher nDCG in nearly all configurations, demonstrating superior retrieval performance. Take the results on ReasonIR-8B for instance, the nDCG improvement reached 22.11% for DeepSeek-R1 and 18.86% for Qwen3-32B. Concurrently, the method substantially reduced computational costs, with savings typically ranging from 15% to 37%. Across 5 dense retrievers and 7 LLM Reasoners, AdaQR achieves an average performance improvement of 7.24% while reducing costs by an average of 28.12%. This demonstrates that AdaQR achieves superior retrieval performance at a lower computational cost.

Different dense retrievers have a significant impact on AdaQR. For ReasonIR-8B, AdaQR delivers the most outstanding and stable performance improvement, achieving an average cost reduction of approximately 29.21% while improving nDCG performance by about 13.18%. In contrast, Qwen3-Embedding-0.6B and Qwen3-Embedding-4B show poor performance. When using these models, AdaQR delivers very limited performance gains and even exhibits slight performance decrements in some few scenarios. This phenomenon aligns with the MRL results in Section 2.2. Embedding transformation for dense retrievers with high MRL values, such as ReasonIR-8B, can be more effectively learned, thus significantly enhancing AdaQR's performance.

AdaQR demonstrates strong generalization across LLM Reasoners and dense retrievers AdaQR offers positive effects across 5 dense retrievers and 7 LLM Reasoners, proving itself as a broadly applicable method for LLM reasoning in text retrieval.

The results presented here are from 7 representative LLM Reasoners, while the results of remaning LLM Reasoners are reported in Appendix A.4.

Rewriting Methods	StackExchange Bio.	Earth.	Econ.	Psy.	Rob.	Stack.	Sus.	Coding Leet.	Pony	AoPS	Theorem-based TheoQ.	TheoT.
Dense Retriever: **BGE-Large** *LLM Reasoner*: **DeepSeek-R1**												
LLM Reasoner	**46.0**	**39.0**	**24.9**	**16.8**	**8.7**	**12.6**	**18.5**	18.5	**0.6**	0.3	**18.7**	**9.9**
Dense Reasoner	20.1	29.6	18.6	6.1	4.6	11.0	13.8	**28.3**	0.2	**8.9**	9.3	1.0
Dense Retriever: **Qwen3-Embedding-4B** *LLM Reasoner*: **Mixtral-8x7B**												
LLM Reasoner	**37.5**	**18.1**	**21.0**	**20.4**	**14.7**	**12.6**	**16.8**	25.5	**2.0**	2.0	22.3	**25.7**
Dense Reasoner	28.1	10.6	14.7	14.6	6.8	11.3	12.0	**46.1**	0.8	**5.8**	**39.1**	24.5
Dense Retriever: **ReasonIR-8B** *LLM Reasoner*: **GLM-Z1**												
LLM Reasoner	**58.2**	**32.8**	**26.5**	**26.6**	**22.7**	**24.7**	**20.6**	25.2	8.3	3.9	**38.5**	**36.4**
Dense Reasoner	48.0	28.9	19.0	21.1	18.2	17.1	14.1	**38.1**	**10.4**	**7.6**	23.6	23.7

Table 1: Performance across 3 domains with 12 tasks: Biology (Bio.), Earth Science (Earth.), Economics (Econ.), Psychology (Psy.), Robotics (Rob.), Stack Overflow (Stack.), Sustainable Living (Sus.), LeetCode (Leet.), Pony, AoPS, TheoremQA with question retrieval (TheoQ.) and with theorem retrieval (TheoT.).

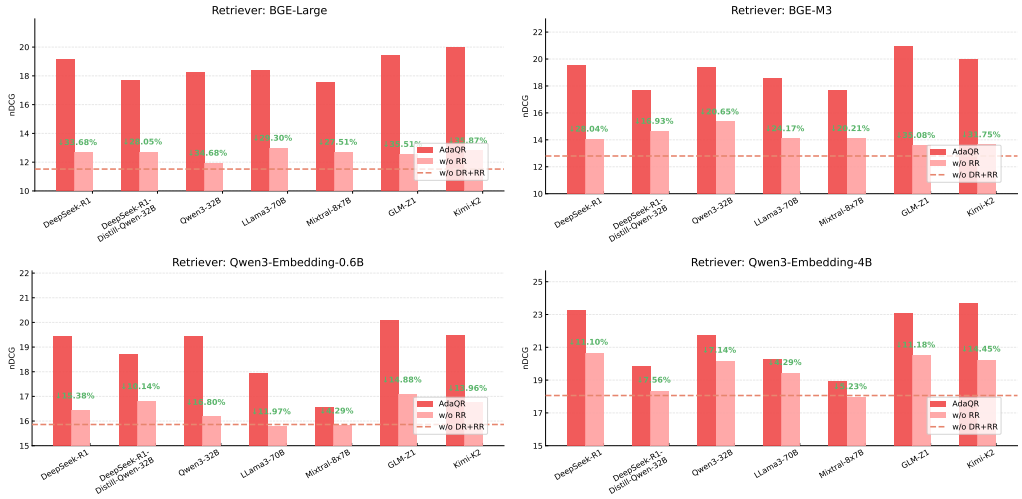

Figure 5: Ablation results after removing the Dense Reasoner and Reasoner Router components.

3.3 Domain and Task Specializations of Reasoners

To further explore the performance of LLM Reasoner and Dense Reasoner on different domains and tasks, we conduct a systematic analysis across the BRIGHT benchmark, including 3 domains (StackExchange, coding and theorem-based) with 12 tasks. To ensure generality, we evaluate performance across several combinations of dense retrievers and LLM Reasoners. The result, presented in Table 1, reveals the difference across datasets.

LLM Reasoners perform better in the StackExchange domain. Compared with Dense Reasoners, LLM Reasoner generally show the best retrieval quality on the StackExchange domain. It demonstrates that for queries which often possesse distinctive linguistic styles and conversational conventions, LLM Reasoner can capture important surface-level or pragmatic cues, whereas Dense Reasoner struggles to fully learn these cues.

Dense Reasoner excels in coding and theorem-based domains. For LeetCode dataset, Dense Reasoner achieves an average nDCG of 37.5%, while LLM Reasoner only achieves 23.06%. Dense Reasoner rewriting based on embedding transformation appears to benefit from this highly structured, formalized language and task-specific terminology, which contributes most of AdaQR's outstanding performance on the full benchmark.

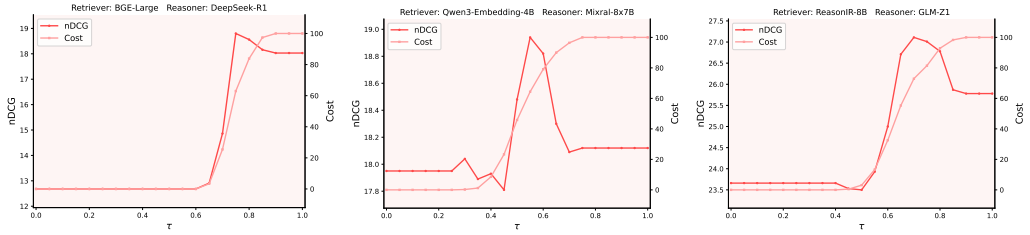

Figure 6: Performance variation of AdaQR under different values of τ.

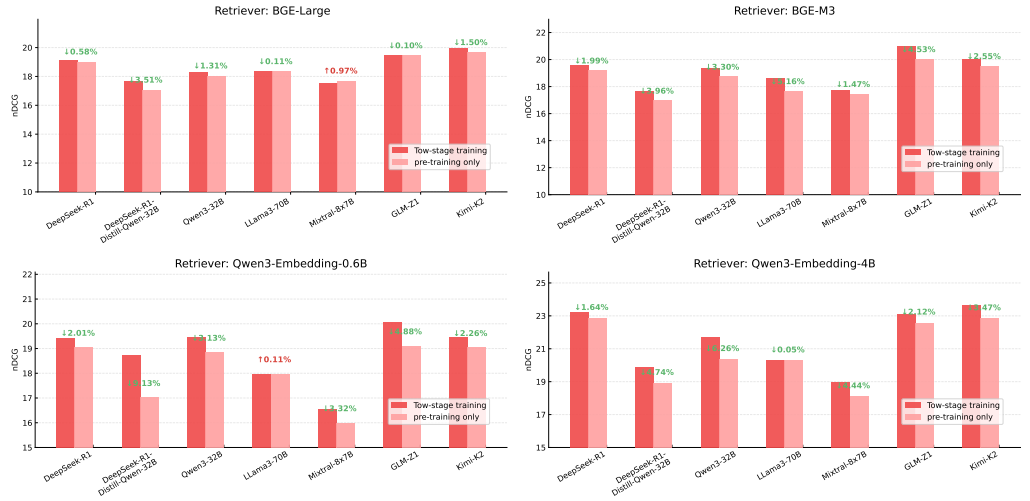

Figure 7: Performance of AdaQR under different training strategies.

3.4 Ablation Study

To explore the individual contributions of the Dense Reasoner and the Reasoner Router, two core components within our proposed AdaQR framework, we conduct a comprehensive ablation study. Specifically, we remove Reasoner Router component, and further removed both Dense Reasoner and Reasoner Router (directly using the original query without any reasoning). Figure 5 reports the results of our ablation study: the average nDCG of AdaQR, w/o RR and w/o DR+RR are 21.05, 17.37 and 15.59 respectively, which clearly reveal the necessity of both components.

3.5 Effect of Decision Threshold τ

We investigate the effect of parameter τ (varying from 0 to 1 with a step size of 0.05), comparing performance and cost in Figure 6. When $\tau = 0$ corresponds to using the Dense Reasoner exclusively, and $\tau = 1$ corresponds to using the LLM Reasoner exclusively. The results indicate that **as τ increases, the cost increases accordingly, while nDCG exhibits a trend of increasing initially and then decreasing**. Concretely, as τ increasing, the Reasoner Router routes more queries to the LLM Reasoner, performance initially improves due to the reasoning capacity of LLMs. However, beyond a threshold (typically around 0.6 to 0.8), Reasoner Router tends to route structured queries that are better predictable by the Dense Reasoner to the LLM, resulting in higher cost and decreasing performance. This observation corroborates the view that τ and oracle anchor serves as a measure of the querys' predictability. AdaQR achieves an adaptive trade-off between performance and cost by selecting τ, balancing effectiveness and efficiency.

Reasoner Router plays a crucial role in AdaQR. Removing Reasoner Router alone leads to a substantial performance drop, suggesting that relying solely on Dense Reasoners cannot achieve both efficiency and effectiveness. When both Reasoner Router and Dense Reasoner are removed, the performance degrades even further, confirming their complementary roles in AdaQR. The Reasoner

Router effectively predicts whether a query is predictable for DR, fully leveraging its low-cost advantage while improving retrieval performance. This dynamic allocation mechanism significantly improves both retrieval quality and efficiency.

It is worth noting that the strategy represented by w/o RR offers **a near-zero-cost query reasoning method** while still outperforming the original query approach, offering a promising option in scenarios with extremely constrained computational resources.

3.6 Impact of Training Strategies

Figure 7 reports the result of the Dense Reasoner under different training strategies, comparing the proposed two-stage strategy with one that omits the fine-tuning stage. Removing the fine-tuning step leads to a slight decline in retrieval performance, with an average decrease of 2.71%. It demonstrates that the fine-tuning phase contributes to dense retrievers adapting for domain distributions, enabling more effective learning of embedding transformation. The slight decline also indicates that the Dense Reasoner has learned effective embedding transformation during the pretraining phase. Our two-stage training strategy enables the Dense Reasoner to first learn the general embedding transformation and then refine that transformation to domain-specific patterns while avoiding forgetting, thereby generating high-quality reasoning embeddings at negligible cost.

4 Related Work

4.1 Efficient Retrieval

Driven by the need to search large corpora with low latency and high throughput, efficient retrieval has been a fundamental and persistent challenge. Efficient text retrieval has been studied in many domains, including using a novel retrieval head (Zhang et al., 2025b), learning the sequential relation between sentences to generate isomorphic embeddings (Zhang et al., 2023), creating a router to assign queries to different expert models (Lee et al., 2025), achieving end-to-end information retrieval with a single LLM (Tang et al., 2024), decomposing the input query into sub-queries to parallelize retrieval process (Zhao et al., 2025), completely removing the deep modeling of queries to maximize retrieval speed (Ma et al., 2025), and avoiding or reducing backfilling to alleviate the performance impact when switching to a new model (Ramanujan et al., 2022; Jaeckle et al., 2023).

4.2 Query Rewriting with LLMs

Query rewriting serves as a crucial preprocessing step, transforming short or ambiguous input queries into well-formed and optimized queries. Recent methods often leverage LLMs to enhance query understanding and expansion. For instance, Hyde, LLM4CS, and query2doc generate pseudo-documents or pseudo-responses to enrich the query context (Gao et al., 2023; Mao et al., 2023; Wang et al., 2023). QA-Expand further explores multi-agent collaboration (Seo & Lee, 2025), while InteR implements iterative information refinement between retrieval models and LLMs (Feng et al., 2024). To improve conversational search, CHIQ proposes five distinct LLM-based rewriting strategies and integrates them (Mo et al., 2024). With the continuous advancement of LLM reasoning capabilities, recently proposed Large Reasoning Models have emerged as state-of-the-art approaches for query rewriting (DeepSeek-AI et al., 2025; Yang et al., 2025; Qin et al., 2025).

5 Conclusion and Future Work

We presented AdaQR, a hybrid query reasoning framework that combines the Dense Reasoner and Reasoner Router to achieve a balance between efficiency and retrieval quality. The Dense Reasoner efficiently approximates LLM reasoning in the embedding space, while the Reasoner Router adaptively directs queries to ensure robustness on challenging cases. Experiments on large-scale retrieval benchmarks show that AdaQR maintains or improves retrieval performance compared to full LLM rewrites, while significantly reducing computation. Future work includes enhancing dense reasoning strategies, extending AdaQR to multi-modal retrieval, refining the routing mechanism, and evaluating its performance in real-world high-throughput systems.

Ethics Statement

This work adheres to the ICLR Code of Ethics. Our study focuses on improving efficiency and effectiveness in information retrieval systems and does not involve human subjects or personally identifiable information. All datasets used are publicly available benchmark datasets, and we comply with their respective licenses and usage guidelines. The proposed AdaQR framework is intended to enhance retrieval performance and reduce computational cost, and we do not foresee any direct harmful societal impacts arising from its use. Potential ethical considerations include bias in retrieval outcomes due to the underlying pre-trained language models or dataset distributions. We encourage users to be aware of such biases when deploying retrieval systems in sensitive applications. We also ensure transparency in our experimental methodology, including dataset usage, model configurations, and evaluation protocols, to support reproducibility and research integrity. Finally, no conflicts of interest or external sponsorships influenced the work reported in this paper.

Reproducibility Statement

We have made every effort to ensure the reproducibility of our results. All datasets used in our experiments are publicly available benchmark datasets, and detailed descriptions of dataset processing, splits, and evaluation metrics are provided in the main paper and supplementary materials. The implementation details of the AdaQR framework, including the Dense Reasoner and Reasoner Router components, as well as hyperparameters, training procedures, and model architectures, are documented in the article.

References

Jun Bai, Chuantao Yin, Zimeng Wu, Jianfei Zhang, Yanmeng Wang, Guanyi Jia, Wenge Rong, and Zhang Xiong. Improving biomedical reqa with consistent nli-transfer and post-whitening. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 20(3):1864–1875, 2022.

Jun Bai, Zhuofan Chen, Zhenzi Li, Hanhua Hong, Jianfei Zhang, Chen Li, Chenghua Lin, and Wenge Rong. Leveraging estimated transferability over human intuition for model selection in text ranking. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 12356–12374, 2024.

Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, Zhuofu Chen, Jialei Cui, Hao Ding, Mengnan Dong, Angang Du, Chenzhuang Du, Dikang Du, Yulun Du, Yu Fan, Yichen Feng, Kelin Fu, Bofei Gao, Hongcheng Gao, Peizhong Gao, Tong Gao, Xinran Gu, Longyu Guan, Haiqing Guo, Jianhang Guo, Hao Hu, Xiaoru Hao, Tianhong He, Weiran He, Wenyang He, Chao Hong, Yangyang Hu, Zhenxing Hu, Weixiao Huang, Zhiqi Huang, Zihao Huang, Tao Jiang, Zhejun Jiang, Xinyi Jin, Yongsheng Kang, Guokun Lai, Cheng Li, Fang Li, Haoyang Li, Ming Li, Wentao Li, Yanhao Li, Yiwei Li, Zhaowei Li, Zheming Li, Hongzhan Lin, Xiaohan Lin, Zongyu Lin, Chengyin Liu, Chenyu Liu, Hongzhang Liu, Jingyuan Liu, Junqi Liu, Liang Liu, Shaowei Liu, T. Y. Liu, Tianwei Liu, Weizhou Liu, Yangyang Liu, Yibo Liu, Yiping Liu, Yue Liu, Zhengying Liu, Enzhe Lu, Lijun Lu, Shengling Ma, Xinyu Ma, Yingwei Ma, Shaoguang Mao, Jie Mei, Xin Men, Yibo Miao, Siyuan Pan, Yebo Peng, Ruoyu Qin, Bowen Qu, Zeyu Shang, Lidong Shi, Shengyuan Shi, Feifan Song, Jianlin Su, Zhengyuan Su, Xinjie Sun, Flood Sung, Heyi Tang, Jiawen Tao, Qifeng Teng, Chensi Wang, Dinglu Wang, Feng Wang, and Haiming Wang. Kimi K2: open agentic intelligence. *CoRR*, abs/2507.20534, 2025.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. BGE m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *CoRR*, abs/2402.03216, 2024a.

Liyang Chen, Yujun Cai, Jieqiong Dong, and Yiwei Wang. BRIGHT+: upgrading the BRIGHT benchmark with marcus, a multi-agent RAG clean-up suite. *CoRR*, abs/2506.07116, 2025a.

Shuhao Chen, Weisen Jiang, Baijiong Lin, James T. Kwok, and Yu Zhang. Routerdc: Query-based router by dual contrastive learning for assembling large language models. In *Advances in Neu-*

ral Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, 2024b.

Xinghao Chen, Anhao Zhao, Heming Xia, Xuan Lu, Hanlin Wang, Yanjun Chen, Wei Zhang, Jian Wang, Wenjie Li, and Xiaoyu Shen. Reasoning beyond language: A comprehensive survey on latent chain-of-thought reasoning. *CoRR*, abs/2505.16782, 2025b.

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, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, and Wangding Zeng. Deepseek-v3 technical report. *CoRR*, abs/2412.19437, 2024.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, and S. S. Li. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948, 2025.

Sriram Dharwada, Himanshu Devrani, Jayant R. Haritsa, and Harish Doraiswamy. Query rewriting via llms. *CoRR*, abs/2502.12918, 2025.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. The llama 3 herd of models. *CoRR*, abs/2407.21783, 2024.

Jiazhan Feng, Chongyang Tao, Xiubo Geng, Tao Shen, Can Xu, Guodong Long, Dongyan Zhao, and Daxin Jiang. Synergistic interplay between search and large language models for information retrieval. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9571–9583, 2024.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1762–1777, 2023.

Florian Jaeckle, Fartash Faghri, Ali Farhadi, Oncel Tuzel, and Hadi Pouransari. Fastfill: Efficient compatible model update. In *The Eleventh International Conference on Learning Representations*, 2023.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. *CoRR*, abs/2310.06825, 2023.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts. *CoRR*, abs/2401.04088, 2024.

Bowen Jin, Jinsung Yoon, Zhen Qin, Ziqi Wang, Wei Xiong, Yu Meng, Jiawei Han, and Sercan Ö. Arik. LLM alignment as retriever optimization: An information retrieval perspective. *CoRR*, abs/2502.03699, 2025.

Ivica Kostric and Krisztian Balog. A surprisingly simple yet effective multi-query rewriting method for conversational passage retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2271–2275, 2024.

Rade Kutil. Biased and unbiased estimation of the circular mean resultant length and its variance. *Statistics*, 46(4):549–561, 2012.

Hyunji Lee, Luca Soldaini, Arman Cohan, Minjoon Seo, and Kyle Lo. Routerretriever: Routing over a mixture of expert embedding models. In *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence*, pp. 11995–12003, 2025.

Hengli Li, Chenxi Li, Tong Wu, Xuekai Zhu, Yuxuan Wang, Zhaoxin Yu, Eric Hanchen Jiang, Song-Chun Zhu, Zixia Jia, Ying Nian Wu, and Zilong Zheng. Seek in the dark: Reasoning via test-time instance-level policy gradient in latent space. *CoRR*, abs/2505.13308, 2025.

Zhengkai Lin, Zhihang Fu, Ze Chen, Chao Chen, Liang Xie, Wenxiao Wang, Deng Cai, Zheng Wang, and Jieping Ye. Controlling thinking speed in reasoning models. *CoRR*, abs/2507.03704, 2025a.

Ziyong Lin, Haoyi Wu, Shu Wang, Kewei Tu, Zilong Zheng, and Zixia Jia. Look both ways and no sink: Converting llms into text encoders without training. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 22839–22853, 2025b.

Guangyuan Ma, Yongliang Ma, Xuanrui Gou, Zhenpeng Su, Ming Zhou, and Songlin Hu. Lightretriever: A llm-based hybrid retrieval architecture with 1000x faster query inference. *CoRR*, abs/2505.12260, 2025.

Kelong Mao, Zhicheng Dou, Fengran Mo, Jiewen Hou, Haonan Chen, and Hongjin Qian. Large language models know your contextual search intent: A prompting framework for conversational search. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 1211–1225, 2023.

Fengran Mo, Abbas Ghaddar, Kelong Mao, Mehdi Rezagholizadeh, Boxing Chen, Qun Liu, and Jian-Yun Nie. CHIQ: contextual history enhancement for improving query rewriting in conversational search. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 2253–2268, 2024.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. MTEB: massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 2006–2029, 2023.

Duy A. Nguyen, Rishi Kesav Mohan, Van Yang, Pritom Saha Akash, and Kevin Chen-Chuan Chang. Rl-based query rewriting with distilled LLM for online e-commerce systems. *CoRR*, abs/2501.18056, 2025.

Xubo Qin, Jun Bai, Jiaqi Li, Zixia Jia, and Zilong Zheng. Tongsearch-qr: Reinforced query reasoning for retrieval. *CoRR*, abs/2506.11603, 2025.

Vivek Ramanujan, Pavan Kumar Anasosalu Vasu, Ali Farhadi, Oncel Tuzel, and Hadi Pouransari. Forward compatible training for large-scale embedding retrieval systems. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19364–19373, 2022.

Stephen E. Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389, 2009.

Wonduk Seo and Seunghyun Lee. Qa-expand: Multi-question answer generation for enhanced query expansion in information retrieval. *CoRR*, abs/2502.08557, 2025.

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. Reasonir: Training retrievers for reasoning tasks. *CoRR*, abs/2504.20595, 2025.

Xuan Shen, Yizhou Wang, Xiangxi Shi, Yanzhi Wang, Pu Zhao, and Jiuxiang Gu. Efficient reasoning with hidden thinking. *CoRR*, abs/2501.19201, 2025.

Hongjin Su, Howard Yen, Mengzhou Xia, Weijia Shi, Niklas Muennighoff, Han-yu Wang, Haisu Liu, Quan Shi, Zachary S. Siegel, Michael Tang, Ruoxi Sun, Jinsung Yoon, Sercan Ö. Arik, Danqi Chen, and Tao Yu. BRIGHT: A realistic and challenging benchmark for reasoning-intensive retrieval. In *The Thirteenth International Conference on Learning Representations*, 2025.

Qiaoyu Tang, Jiawei Chen, Zhuoqun Li, Bowen Yu, Yaojie Lu, Cheng Fu, Haiyang Yu, Hongyu Lin, Fei Huang, Ben He, Xianpei Han, Le Sun, and Yongbin Li. Self-retrieval: End-to-end information retrieval with one large language model. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024*, 2024.

Liang Wang, Nan Yang, and Furu Wei. Query2doc: Query expansion with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 9414–9423, 2023.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. C-pack: Packaged resources to advance general chinese embedding. *CoRR*, abs/2309.07597, 2023.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jian Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *CoRR*, abs/2505.09388, 2025.

Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Juanzi Li, Lei Zhao, Lindong Wu, Lucen Zhong, Mingdao Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiao Xia, Xiaohan Zhang, Xiaotao Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang, Yifan An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi, Zhaoyu Wang, Zhen Yang, Zhengxiao Du, Zhenyu Hou, and Zihan Wang. Chatglm: A family of large language models from GLM-130B to GLM-4 all tools. *CoRR*, abs/2406.12793, 2024.

Shunyu Zhang, Yaobo Liang, Ming Gong, Daxin Jiang, and Nan Duan. Modeling sequential sentence relation to improve cross-lingual dense retrieval. In *The Eleventh International Conference on Learning Representations*, 2023.

Tuo Zhang, Asal Mehradfar, Dimitrios Dimitriadis, and Salman Avestimehr. Leveraging uncertainty estimation for efficient LLM routing. *CoRR*, abs/2502.11021, 2025a.

Wuwei Zhang, Fangcong Yin, Howard Yen, Danqi Chen, and Xi Ye. Query-focused retrieval heads improve long-context reasoning and re-ranking. *CoRR*, abs/2506.09944, 2025b.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *CoRR*, abs/2506.05176, 2025c.

Yiqun Zhang, Hao Li, Chenxu Wang, Linyao Chen, Qiaosheng Zhang, Peng Ye, Shi Feng, Daling Wang, Zhen Wang, Xinrun Wang, Jia Xu, Lei Bai, Wanli Ouyang, and Shuyue Hu. The avengers: A simple recipe for uniting smaller language models to challenge proprietary giants. *CoRR*, abs/2505.19797, 2025d.

Shu Zhao, Tan Yu, Anbang Xu, Japinder Singh, Aaditya Shukla, and Rama Akkiraju. Parallelsearch: Train your llms to decompose query and search sub-queries in parallel with reinforcement learning. *CoRR*, abs/2508.09303, 2025.

Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. Dense text retrieval based on pretrained language models: A survey. *ACM Transactions on Information Systems*, 42(4):1–60, 2024.

Richard Zhuang, Tianhao Wu, Zhaojin Wen, Andrew Li, Jiantao Jiao, and Kannan Ramchandran. Embedllm: Learning compact representations of large language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

A Appendix

A.1 LLM Usage

In this work, we used ChatGPT (GPT-5) as an assistive tool for drafting and refining text in the introduction and related work. All content produced with the assistance of ChatGPT was reviewed, revised, and verified by the authors. ChatGPT contributed to wording suggestions and phrasing improvements but did not contribute independently to research ideation, experimental design, or result analysis. The authors take full responsibility for all content in this paper.

A.2 Datasets

For StackExchange for Dense Reasoner pre-training, we collect 9795 queries from 17 domains: ai, biology, bioinformatics, chemistry, codereview, computergraphics, cs, earthscience, economics, math, mathoverflow, philosophy, physics, robotics, stackoverflow, softwareengineering, sustainability. Each dataset contributes 600 queries, except for computergraphics (364), philosophy (599) and sustainability (432). During the collection phase, we excluded queries containing images and selected only those whose answers were chosen. We also carefully reviewed all candidate queries to ensure no overlap with queries from the BRIGHT benchmark. The prompt we use for query reasoning is shown in Figure 10.

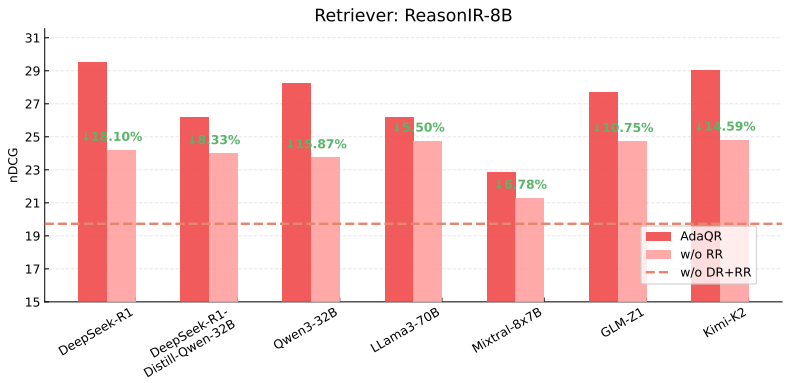

Figure 8: Ablation results on components of ReasonIR-8B

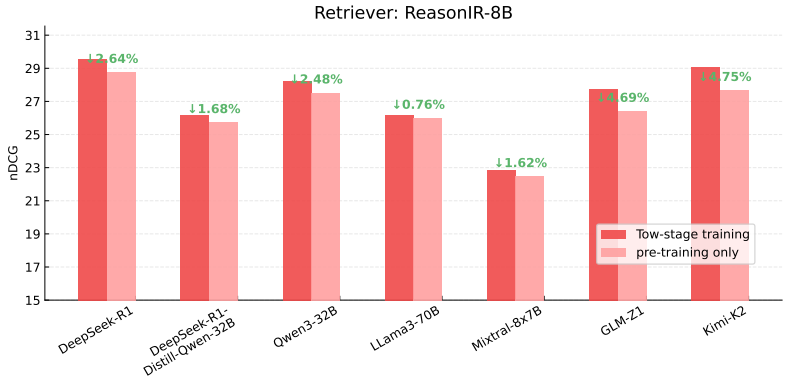

Figure 9: Performance of AdaQR under different training strategies of ReasonIR-8B

A.3 Results of ReasonIR-8B

We present results of ReasonIR-8B in Figure 8 and Frigure 9, which are not presented in Section 3.

A.4 Main results of additional LLMs

Fig 11 reports the retrieval performance improvement and reasoning cost reduction achieved by AdaQR compared to LLM-based reasoning across 10 additional LLMs.

Prompt template for LLM reasoning

```
{Query}
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.
```

Figure 10: Prompt template used to guide the LLM Reasoner to thoroughly analyze and provide detailed explanations for a given query.

Figure 11: Retrieval performance improvement and reasoning cost across 10 additional LLMs