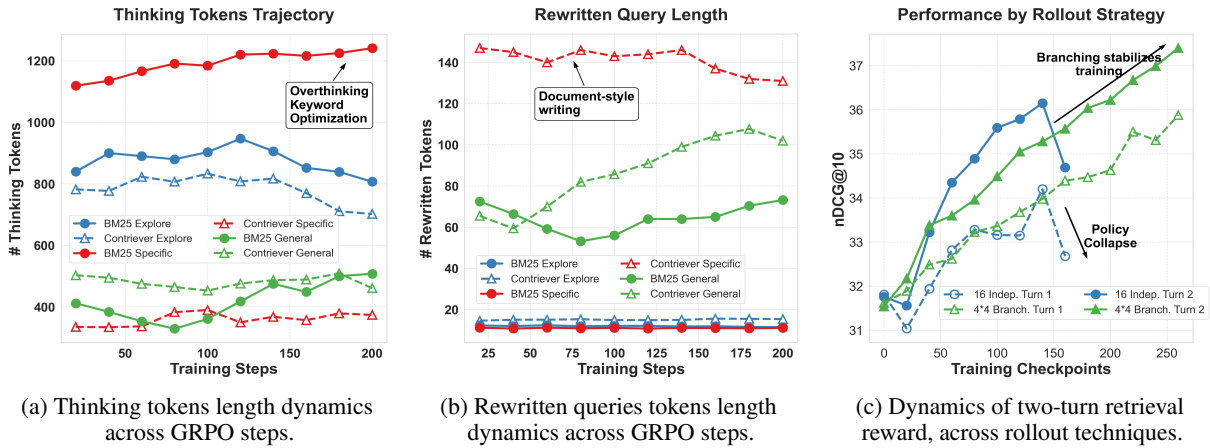


# Understanding the Behaviors of Environment-aware Information Retrieval

Ruifeng Yuan<sup>1,2,3\*</sup> Chaohao Yuan<sup>2,4\*</sup> David Dai<sup>5\*</sup> Yu Rong<sup>2,3</sup> Hong Cheng<sup>4</sup>  
 Hou Pong Chan<sup>6†</sup> Chenghao Xiao<sup>7†</sup>

<sup>1</sup>Fudan University <sup>2</sup>Alibaba Group <sup>3</sup>Hupan Lab <sup>4</sup>CUHK  
<sup>5</sup>Stanford <sup>6</sup>University of Macau <sup>7</sup>SUFE



(a) Thinking tokens length dynamics across GRPO steps.

(b) Rewritten queries tokens length dynamics across GRPO steps.

(c) Dynamics of two-turn retrieval reward, across rollout techniques.

**User Question:** Retrieve a Wikipedia article that answers: Who wrote God's Plan? The retriever you will be using is Contriever.  
**Assistant:** <think>{... Omitted...} Since Contriever uses document-to-document retrieval based on spans from Wikipedia, I should create a document that includes key information about the song. I need to make sure the document includes {... Omitted...} Putting it all together, the hypothetical document should have the song title as the main heading, followed by the artist, release year, genre, writer, and a brief note on its success. </think> <rewrite>"God's Plan" was written and produced by Canadian rapper and songwriter Drake. Released in 2018 as the lead single from his fifth studio album "Scorpion", the track features a minimalist production with a repetitive piano loop and a soulful vocal sample. "God's Plan" received widespread acclaim for its introspective lyrics and became a commercial success, reaching number one on the Billboard Hot 100 chart. </rewrite>

(d) Query rewriter learns to rewrite queries as document-style which Contriever favors.

**User Question:** who does dwyane wade play for right now? You will be using bm25 to retrieve.  
**Assistant:** <think>{... Omitted...} First, I need to figure out the best way to phrase this query for BM25. BM25 is a keyword-based retrieval system, so including relevant keywords is crucial. The original query is "who does dwyane wade play for right now." The key elements here are "Dwyane Wade," "play for," and "right now." Since BM25 relies on matching keywords in the document, I should make sure these terms are included. However, "right now" might be a bit too specific or could be interpreted as a time-related keyword. Maybe replacing "right now" with "current" or "now" could be better. Also, using "current team" instead of "play for right now" might be more effective. {... Omitted...} Between these, "Dwyane Wade current team" is more likely to yield the correct results because it's a common way to phrase such questions. <rewrite>Dwyane Wade current team </rewrite>

(e) For bm25, the query rewriter learns to rewrite queries as keywords-style.

**User Question:** Jim Price appeared on the Rolling Stones Album released in what month?  
**Assistant:** {... Omitted...}<rewrite>{... Omitted...}</rewrite> Retrieved Doc: {... Omitted...}  
**Assistant:** {... Omitted...} But looking at the top documents, doc0 mentions Jim Price touring with The Rolling Stones from 1970-1973 and appearing on "Sticky Fingers", "Exile on Main St.", and "Goats Head Soup". Wait, in the initial retrieval, the first query might have assumed "Steel Train" was released in May, but maybe that's not confirmed in all sources. {... Omitted...}<rewrite2>Jim Price was a horn session player for The Rolling Stones, appearing on their albums "Sticky Fingers," "Exile on Main St.," and "Goats Head Soup" during his tenure from 1970 to 1973. To determine the exact month of release for the Rolling Stones album that featured Jim Price, it is necessary to identify which of these albums were released in a specific month.</rewrite2>

(f) Two-turn environment-aware retrieval.

Figure 1: (a-c) GRPO learning Dynamics across retrievers and training strategies. (d-f) Case studies.

## Abstract

Recent retrieval-augmented generation (RAG) approaches have demonstrated strong capability in handling complex queries, yet current research overlooks a critical challenge: different retrievers require fundamentally different query formulation strategies for optimal performance. In this work, we present the first systematic analysis of how LLMs can learn

to adapt their query formulation strategies for different retrievers via reinforcement learning (RL). Our empirical study reveals that RL effectively teaches an LLM to tailor its queries to specific retriever characteristics. We discover that different retrievers exhibit surprisingly distinct optimal query styles (e.g., descriptive vs. question-like), suggesting strategies learned for one retriever ineffective for another. We further show that performance can be enhanced by incorporating retriever-specific human guidance and by scaling model size. To facilitate

\*Equal Contribution.

†Corresponding authors, [kenchanhp@gmail.com](mailto:kenchanhp@gmail.com), [justinchenghaoxiao@gmail.com](mailto:justinchenghaoxiao@gmail.com)

learning over multi-retrieval-step trajectories, we introduce a branching-based rollout technique that improves training stability. Our work provides the first empirical evidence and actionable insights for building truly retriever-aware RAG systems. Code and resources are available at [LCO-Embedding/Envs-aware-Information-Retrieval](#).

## 1 Introduction

Retrieval-augmented generation (RAG) is a standard approach for augmenting the knowledge of large language models (LLMs) by generating queries and retrieving relevant information from external sources. Recently, Agentic RAG has emerged as a more sophisticated paradigm that, rather than performing a single retrieval step, decomposes complex queries and iteratively executes multiple retrieval operations to fulfill complex task requirements, such as generating comprehensive research reports in real-world applications.

Most agentic RAG approaches treat retrieval as a uniform “tool call,” operating under the assumption that it occurs exclusively through a single commercial search engine (e.g., Google) and optimizing for that specific engine. However, retrieval from locally stored information repositories remains critically important in real-world applications, including private financial reports and similar patient case databases. In practice, agentic RAG systems must accommodate diverse retrieval backends. Different retrievers may require distinct optimal query formulation strategies to maximize retrieval effectiveness—an LLM should fundamentally adapt how it formulates queries depending on which retriever it is querying. However, current research lacks adequate investigation into how LLMs can develop retriever-aware query formulation capabilities.

In this work, we present **the first systematic analysis of how LLMs learn to adapt their query formulation strategies for different retrievers via reinforcement learning (RL)**. We systematically examine the effects of different retriever types, the comprehensiveness of human guidance in prompts, and the scaling of model size and retrieval steps, via the following research questions.

**RQ1: Can LLMs adapt their query formulation strategies to local retrievers via RL? If so, are strategies learned for one retriever transferable to others?** We find that RL effectively adapts an LLM’s query formulation strategy to local retrievers. Moreover, we discover that different

retrievers exhibit distinct optimal query formulation strategies. For example, Contriever (Izacard et al., 2022) requires long-form descriptive queries while Qwen3-Embedding (Zhang et al., 2025) performs better with question-like queries. Consequently, *strategies optimized for one retriever cannot be effectively transferred to different retrievers*.

**RQ2: How does the comprehensiveness of human guidance in prompts affect the learning of query formulation?** We design three levels of knowledge comprehensiveness in human guidance: from general query rewriting strategies to retriever-specific query formulation by human experts. We find that more comprehensive retriever-specific knowledge in prompts leads to better LLM performance, suggesting that LLMs benefit from human expertise to guide RL exploration in query formulation.

**RQ3: Can we enhance query formulation performance by scaling up model size and the number of retrieval steps?** We find that larger models achieve better performance and can even discover policies that human experts do not anticipate. Moreover, to support multi-step retrieval trajectories during rollouts, we introduce a branching technique that isolates the learning signal to the final turn. Experimental results show that increasing the number of retrieval steps enhances query formulation capability, as the model can leverage documents retrieved in earlier steps to refine subsequent query formulation strategies.

We summarize our key contribution as follows: (1) We present the first systematic empirical study analyzing how LLMs learn to adapt their query formulation strategies for different retrievers via reinforcement learning. (2) Our results reveal distinct optimal query formulation strategies for prominent retrievers, providing actionable insights for RAG system practitioners. (3) We introduce a branching-based optimization technique to support multi-step retrieval trajectory rollouts, facilitating stable RL training for multi-step query strategy optimization.

## 2 Methodology

We formulate retriever-aware query adaptation as a reinforcement learning (RL) problem, in which an LLM-based query rewriter policy  $\pi_\theta$  learns to generate optimal search queries  $q$  for a given black-box retriever environment  $\mathcal{E}$ , conditioned on a user-issued question. The policy is optimized using Group Relative Policy Optimization (GRPO) (Shao

et al., 2024), which enables stable learning without requiring a value-function critic.

## 2.1 Reward Function Formulation

A critical component of our framework is the reward signal provided by the environment. For a generated query  $q$ , the retriever returns a set of documents  $D$ . We define the reward  $r(q, D)$  as the **Normalized Discounted Cumulative Gain (nDCG@10)** of the retrieved documents  $D$  with respect to the ground-truth passage  $d^*$ . We select nDCG over simple recall to incentivize the model to not only find the correct document but to rank it highly, mirroring real-world RAG requirements.

## 2.2 Scenario 1: Single-Turn Retrieval

In the single-turn setting, the policy generates a rewrite  $q$  given an input  $x$ . For each input, we sample a group of  $G$  outputs  $\{q_1, \dots, q_G\}$  from the policy  $\pi_{\theta_{old}}$ . The GRPO objective is maximizing:

$$\mathcal{L}_{GRPO}(\theta) = -\frac{1}{G} \sum_{i=1}^G \frac{1}{|y_i|} \sum_{t=1}^{|y_i|} \left[ \min(\rho_{i,t} A_i, \text{clip}(\rho_{i,t}, 1 - \epsilon, 1 + \epsilon) A_i) - \beta \mathbb{D}_{KL}(\pi_{\theta} || \pi_{ref}) \right], \quad (1)$$

where  $\rho_i = \frac{\pi_{\theta}(q_i|x)}{\pi_{\theta_{old}}(q_i|x)}$  is the importance sampling ratio. The advantage  $A_i$  is computed by normalizing the reward  $r_i$  against the group statistics:

$$A_i = \frac{r_i - \mu_{group}}{\sigma_{group} + \delta}, \quad (2)$$

where  $\mu_{group}$  and  $\sigma_{group}$  are the mean and standard deviation of rewards within the group.

## 2.3 Scenario 2: Multi-Turn Iterative Retrieval

To enable the agent to refine queries based on initial feedback, we define a multi-turn trajectory  $\tau_i = [q_{i,1}, e_{i,1}, q_{i,2}, e_{i,2}, \dots, q_{i,K}]$ , where  $q_{i,k}$  is the query (and reasoning) at turn  $k$ , and  $e_{i,k}$  represents the search results (masked during training).

**Weighted Reward Aggregation.** To value the final answer while encouraging useful intermediate steps, we compute the total trajectory reward  $R_i$  as a weighted sum:

$$R_i = \sum_{k=1}^K \lambda_k \cdot r(q_{i,k}, e_{i,k}). \quad (3)$$

In practice, we assign a higher weight to later rounds, which is discussed in implementation details and multi-round training techniques section.

## 2.4 Branching Rollout Strategy

A key challenge in multi-turn RL is the high variance of intermediate rewards, which leads to training collapse. Hence, we introduce a **Branching Rollout** strategy that stabilizes credit assignment.

In this work, we focus on a two-turn RL setting. Instead of independent linear trajectories (e.g.,  $16 \times 1$ ), we implement a branching structure (e.g.,  $4 \times 4$ ) where we sample  $M$  first-turn actions, and for each, branch into  $N$  second-turn completions. This yields two benefits:

- **Monte Carlo Estimation for Turn 1:** The value of a first-turn action  $q_{i,1}$  is estimated by averaging over its  $N$  branches, providing a robust signal for the intermediate step:

$$V(q_{i,1}) \approx r(q_{i,1}) + \frac{1}{N} \sum_{j=1}^N r(q_{i,2}^{(j)}). \quad (4)$$

- **Variance Reduction for Turn 2:** By grouping the  $N$  branches that share the same history, the baseline (mean reward) becomes specific to that sub-group. This isolates the learning signal for the second turn, ensuring that the advantage calculation reflects the quality of the second rewrite independent of the first turn’s quality.

## 3 Experiments

### 3.1 Settings

**In-distribution Evaluation** We leverage RAGBench (Friel et al., 2024), a large-scale benchmark for evaluating retrieval-augmented generation systems. RAGBench comprises more than 100,000 samples, spans a broad range of general and specialized domains, and exhibits substantial variation in document structure and information needs, which makes it well suited for a comprehensive evaluation of model performance. To adapt RAGBench for retrieval tasks, we first aggregate all documents in the dataset into a global retrieval corpus, and treat the questions as search queries and their associate contexts as the positive pair for retrieval. Additionally, for corpus de-duplication, we employ Qwen-Embedding-0.6B (Zhang et al., 2025) to encode the documents, retaining only one instance from pairs that share a similarity greater than 0.99.

**OOD Evaluation** To evaluate the transferability of LLM query rewriter trained on general-domain RAG datasets like RAGBench, we conduct zero-shot experiments to the commonly-used BEIR benchmark (Thakur et al., 2021).

**Domain-specific Transfer** To evaluate how well retriever-aware query rewriters trained on general-domain datasets generalize to challenging domain-specific environments, we conduct an in-depth case study in the financial domain. Although underexplored in existing retrieval benchmarks, finance is an ideal stress test for cross-domain transfer due to (1) its pronounced distribution shift from general-domain corpora induced by domain-specific terminology (e.g., SG&A, currency fluctuation, gross margin), (2) the hybrid structure of financial documents that interleaves text with tabular content, and (3) the practical relevance of information retrieval in real-world workflows in finance. To this end, we construct a financial retrieval environment leveraging FinAgentBench (Choi et al., 2025) and describe the process in detail in Appendix B.

### 3.2 Implementation Details

**Infrastructure** We serve the retrieval environments (retrievers and pre-indexed corpus) as APIs using FastAPI and uvicorn. The retrieval environments are served on different nodes, isolated from the main GRPO training process, which secures no interference with the RL training.

**RL implementation** For multi-turn search, we assign 0.5 weighting to the first-round retrieval and 1 to the second-round retrieval. We apply masking to the search results returned by the retrievers, and only compute the loss of the reasoning and the rewritten queries. This is based on the principle of not punishing or rewarding a model for texts it didn't generate. We further ablate whether it is more optimal to apply  $n^2$  rollouts to the full unconditional two-turn search trajectories, or  $n$  rollouts to the second-turn rewriting conditioned on  $n$  first-turn rewriting and search results.

**Retrievers** We select four representative retrieval systems, including Qwen3-Embedding (Zhang et al., 2025), all-MiniLM-L6-v2 (Sentence-Transformers, 2025), Contriever (Izacard et al., 2022), and BM25 (Robertson and Zaragoza, 2009).

Qwen3-Embedding and all-MiniLM-L6-v2 are dense embedding models trained on large-scale supervised pairs, and they respectively represent decoder/high-capability/large and encoder/low-capability/small models.

Contriever is an unsupervised representation model trained on randomly cropped spans of Wikipedia article as positive pairs. Therefore, Contriever intuitively prefers retrieving documents us-

#### Question-to-Statement Rewriting Policy discovered by a 14B agent under Contriever environment.

User Question: Is it possible to be white and latino?  
**assistant:** <think> Okay, let's tackle this question. The user is asking if it's possible to be both white and Latino. I need to figure out the best way to rewrite the query for Contriever to get the most relevant documents.

(🤔) First, I should understand what Contriever does. It's trained on Wikipedia articles, using two spans as positive pairs. So, the model is good at finding documents where two phrases are related. The original query is about the intersection of race and ethnicity, specifically being white and Latino. {... More verbose reasoning omitted}... To get the best results, the rewritten query should capture the essence of the question. (💡) Instead of a direct question, maybe a statement that reflects the core of the query. For example, "White individuals identifying as Latino" or "Intersection of race and ethnicity in Latino communities." (🤔) But I need to make sure it's a query that Contriever can handle effectively.

Another angle is to use terms that are more likely to appear in Wikipedia articles. Terms like "ethnicity and race intersection," "Latino identity," or "multiracial identity." Maybe combining both terms in a way that Contriever can retrieve documents discussing both aspects.

(🤔💡) Wait, the original question is a yes/no question, but Contriever might perform better with a more descriptive query. Alternatively, using a phrase that's more likely to appear in documents discussing both concepts. Maybe "Latino and white identity intersection" or "Can someone be both white and Latino." {... More verbose reasoning omitted}...

After considering these options, the best approach is to (🏆) rephrase the question into a statement that clearly states the intersection, which is likely to be covered in relevant documents. So, the rewritten query might be "Can a person be both white and Latino?" which is similar to the original but phrased as a statement that Contriever can retrieve documents about. </think>  
 <rewrite> Being white and Latino </rewrite>

(🚀) Result Metric: nDCG@10 of 0.965

Figure 2: Novel Strategy Discovery by rewriting agents. Emojis are annotated with meanings: 🤔: Retriever-aware reflections. 💡: Policy Proposal. 🏆: Policy Confirmation. 🚀: Successful Retrieval which yields advantage in GRPO rollouts.

ing document-style query, while providing surpar performance using question-style query.

BM25 is a sparse retrieval algorithm, which in its nature prefers keyword-style retrieval. When the metadata (e.g., inverse document frequency) is computed on non-massive corpus, it is also sensitive to small-variants of phrasing and spelling (e.g., carryforward vs. carryforwards).

We follow the optimal encoding setting of all retrieval systems, such as using mean pooling for Contriever, and using the query instruction template for Qwen3-Embedding, etc.

Retriever	CovidQA	DelucionQA	EManual	ExpertQA	FinQA	HAGRID	HotpotQA	MS Marco	PubmedQA	TAT-QA	TechQA	Avg.
Contriever	36.0	61.6	55.1	52.5	14.4	68.9	43.8	72.0	62.3	8.6	24.7	45.5
Contriever + rewrite general	40.4	70.0	65.7	60.8	21.0	85.8	64.1	77.2	71.4	13.6	30.0	54.5†
Contriever + rewrite direct	42.1	71.3	71.1	62.6	23.3	87.2	63.9	78.1	72.9	14.9	31.9	56.3†
Contriever + rewrite explore	41.9	<b>72.1</b>	65.2	54.7	18.1	82.4	63.9	77.2	68.5	12.3	27.9	53.2†
Contriever + rewrite general + turn2	41.8	68.7	62.8	<b>64.0</b>	22.9	<b>87.9</b>	<b>68.1</b>	76.0	69.9	13.9	30.9	55.2†
Contriever + rewrite direct + turn2	<b>42.6</b>	70.1	<b>71.5</b>	63.5	<b>23.3</b>	86.1	64.9	<b>78.5</b>	<b>73.0</b>	<b>15.0</b>	<b>32.5</b>	<b>56.5†</b>
BM25	30.8	57.3	40.0	50.6	20.0	67.3	68.7	57.9	48.1	10.4	44.0	45.0
BM25 + rewrite general	37.6	66.1	56.1	49.2	29.3	74.9	71.1	69.8	58.6	21.5	49.7	53.1†
BM25 + rewrite explore	<b>40.5</b>	<b>66.1</b>	<b>66.3</b>	<b>46.1</b>	<b>33.4</b>	<b>77.9</b>	<b>82.6</b>	<b>72.5</b>	<b>61.9</b>	<b>24.9</b>	<b>53.7</b>	<b>56.9†</b>
BM25 + rewrite direct	40.2	66.0	66.2	47.0	31.3	77.4	82.2	71.3	61.7	24.4	50.1	56.2
all-MiniLM-L6-v2	39.7	68.1	65.5	61.2	11.1	84.4	58.8	79.3	72.2	11.2	30.3	52.9
all-MiniLM-L6-v2 + rewrite general	<b>41.4</b>	<b>72.1</b>	<b>71.1</b>	<b>61.2</b>	<b>13.5</b>	<b>86.4</b>	<b>61.1</b>	<b>81.8</b>	<b>75.4</b>	<b>13.7</b>	<b>31.4</b>	<b>55.4†</b>
Qwen3-Embedding-0.6B	56.3	77.8	73.7	66.7	27.1	89.2	67.0	81.0	79.1	17.6	41.2	61.5
Qwen3-Embedding-0.6B + rewrite general	<b>60.5</b>	<b>83.4</b>	<b>77.4</b>	<b>66.9</b>	<b>30.0</b>	<b>89.4</b>	<b>68.0</b>	<b>84.5</b>	<b>83.2</b>	<b>20.1</b>	<b>41.2</b>	<b>64.1†</b>

Table 1: Results of LLM query rewriters with different retrievers on the RAGBench.

Retriever	ArguAna	MSMARCO	TREC-COVID	NFCorpus	NQ	HotpotQA	FiQA	Touché	Quora	DBPedia	SCIDOCS	FEVER	Climate-FEVER	SciFact	Avg.
Contriever	33.55	36.62	17.36	27.11	18.05	41.01	12.41	6.75	83.36	25.08	10.97	27.22	7.16	57.14	28.84
w/ Our rewriter	32.75	<b>49.23†</b>	<b>32.08†</b>	26.71	<b>36.28†</b>	<b>52.44†</b>	<b>16.17†</b>	<b>7.68†</b>	77.27	<b>32.18†</b>	<b>13.10†</b>	<b>37.52†</b>	<b>11.75†</b>	<b>64.54†</b>	<b>34.98†</b>
w/ Gemini-2.5-flash															
- general prompt	31.21	38.34	28.21	26.58	16.36	39.16	10.33	4.58	68.99	23.38	10.55	22.34	6.21	51.31	26.97
- specific Contriever prompt	30.72	37.08	28.13	26.25	18.70	39.40	10.13	3.85	64.87	25.70	10.79	24.77	6.47	56.70	27.40

Table 2: Zero-shot Adaptation of LLM query rewriters to the BEIR benchmark.

Retriever	FinAgentBench
Contriever	6.43
Contriever + rewrite	<b>7.39†</b>
BM25	8.17
BM25 + rewrite	<b>9.02†</b>
all-MiniLM-L6-v2	6.69
all-MiniLM-L6-v2 + rewrite	<b>7.17†</b>
Qwen3-Embedding-0.6B	9.10
Qwen3-Embedding-0.6B + rewrite	<b>10.81†</b>

Table 3: nDCG@10 performance on FinAgentBench.

## 4 Results

### 4.1 Main Results

**RQ1.1: Can LLMs adapt their query formulation strategies to local retrievers via RL?** We observe significant performance gains across diverse retrievers on the in-distribution RAGBench dataset (Table 1). Adapting query formulation to the specific retriever—whether through general prompts, specific instructions, or exploratory RL—consistently outperforms the baseline usage of the retriever alone, e.g., the BM25 retriever sees a substantial jump in nDCG@10 when using the “explore” rewriting strategy compared to its raw baseline. This confirms that the model successfully learns distinct “dialects” suited for different backends.

The learned query rewriter LLM agents also demonstrate strong generalization capabilities when applied to out-of-distribution and domain-specific environments. On the BEIR benchmark (Table 2), our query rewriter improves the average performance of Contriever from 28.84 to 34.98 in

a zero-shot setting. In contrast, Gemini-2.5-Flash fails to provide useful rewritten queries even if under the specific prompts aligned with our rewriters, resulting in performance drop.

Furthermore, results on the specialized FinAgentBench (Table 3) show universal improvements across all tested systems—including Contriever, BM25, and Qwen3-Embedding—proving that the learned structural adaptations remain effective even under significant domain shifts.

Although our primary focus is on isolating the retrieval component from the full RAG pipeline, improved retrieval quality directly increases the likelihood that the gold document ( $d^*$ ) is ranked highly, thereby improving downstream answer accuracy. In Appendix D, we empirically verify that these retrieval gains translate into improvements in end-to-end RAG performance.

### 4.2 Measuring the Discrepancy of Optimal Queries

**RQ1.2: Can strategies learned for one retriever transferable to others?** To quantify the transferability of learned query formulation strategies across retrievers, we introduce **Retrieval Environment MMD (RE-MMD)**, a metric based on Maximum Mean Discrepancy that measures the distance between the distributions of optimal queries in a source retrieval environment  $\mathcal{E}_{\text{src}}$  and a target environment  $\mathcal{E}_{\text{tgt}}$ . We decompose distributional discrepancies into two complementary types of drift:

- **Semantic Drift (Intent).** Do the queries target different underlying information needs? We measure this using a dense semantic encoder  $\phi_{\text{sem}}$

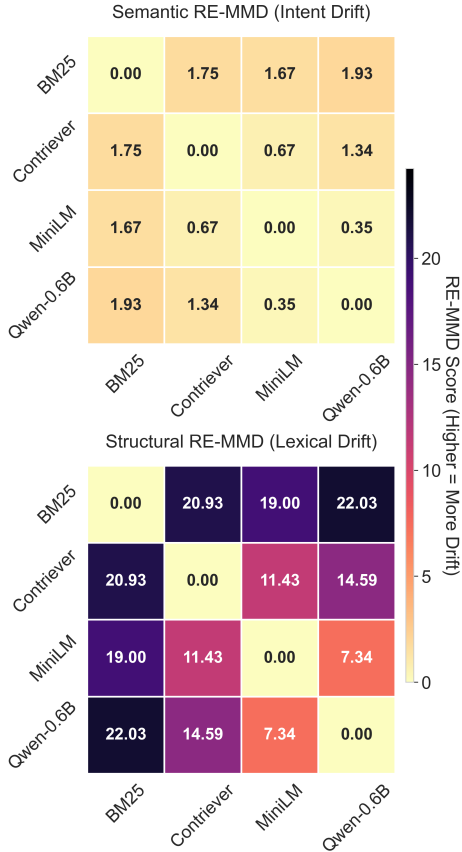


Figure 3: Measuring semantic and structural drifts of optimal queries across retrieval environments. Scores are our proposed RE-MMD values multiplied by 100.

(e.g., MiniLM) that maps queries into an intent representation space.

- **Structural Drift (Formulation).** Do the queries differ in their surface-level formulation or “dialect” (e.g., keyword-based versus natural-language expressions)? We capture this using lexical representations  $\phi_{\text{struct}}$ .

Formally, we define the squared RE-MMD distance in a Reproducing Kernel Hilbert Space (RKHS)  $\mathcal{H}$  as:  $\text{RE-MMD}^2(\mathcal{E}_{\text{src}}, \mathcal{E}_{\text{tgt}}) = \|\mu_{\text{src}} - \mu_{\text{tgt}}\|_{\mathcal{H}}^2$ , where  $\mu = \mathbb{E}_{q \sim \mathcal{P}}[\phi(q)]$  denotes the expected embedding of successful queries in corresponding environments. Details of the empirical estimation using kernel-based methods are in Appendix C.

**Results.** Figure 3 reports the RE-MMD scores (scaled by 100) across retriever pairs. We observe a clear contrast between the two types of drift. **Semantic RE-MMD remains consistently low** (typically  $< 2.0$ ), indicating that the underlying information need is largely preserved across retrievers. In contrast, **Structural RE-MMD is an order of**

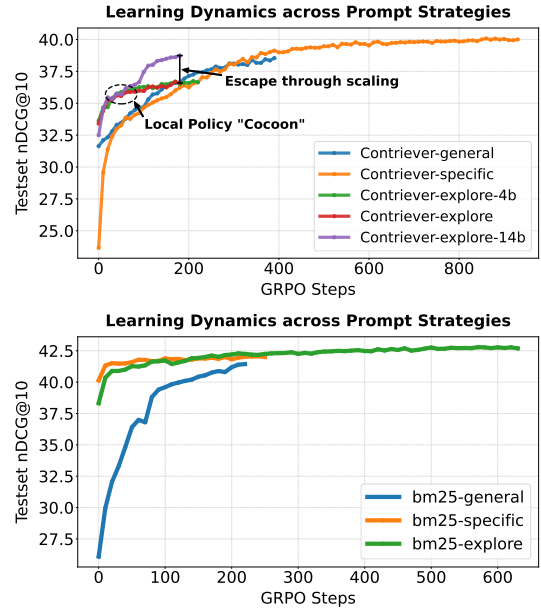


Figure 4: Learning dynamics across different prompt methods and model sizes of query rewriter LLM.

**magnitude larger** (often  $> 20.0$ ), especially between sparse retrievers (e.g., BM25) and dense retrievers (e.g., Contriever). These results provide quantitative evidence that strategy transfer failures are primarily *stylistic* rather than *intentional*: the agent fails not because it seeks different information, but because it cannot effectively “code-switch” into the query formulation style—such as keyword density versus declarative phrasing—required by the target retrieval environment. Overall, *our findings suggest that query formulation policies optimized for one retriever generally do not transfer reliably to others.*

### 4.3 Human Guidance in Prompt

**RQ2: How does the comprehensiveness of human guidance in prompts affect the learning of query formulation?** Prompt design is known to substantially influence the learning dynamics of GRPO-based optimization. In this section, we investigate how different levels of human guidance embedded in prompts shape the learning behavior of query formulation policies. Specifically, we analyze learning dynamics under different prompt priors that vary in the amount of retriever-specific knowledge provided to the model.

We conduct in-depth experiments in retrieval environments built with Contriever and bm25, which represent two retriever classes with distinct and human-interpretable optimal query behaviors.

Owing to its document span, document span contrastive training objective, `Contriever` is expected to perform best when the original query is transformed into a hypothetical document resembling the target document. In contrast, the statistical nature of `bm25` inherently favors concise, keyword-oriented queries.

We design three levels of prompt comprehensively (Figure 6 in Appendix F provides the prompts in detail):

- **General Prompts.** We provide general query rewriting guidelines along with five candidate rewriting strategies. The query rewriter is instructed to reason over these options and select one strategy per rollout. In this setting, the rewriter is not informed of the underlying retriever and must infer its behavior solely through GRPO reward signals during training.
- **Exploratory Prompts.** We inform the query rewriter of the retriever being used and provide a high-level description of how it was trained. The rewriter is encouraged to explore and infer an optimal query formulation strategy based on its understanding of the retriever’s behavior.
- **Specific Prompts.** We explicitly inform the query rewriter of the retriever and prescribe a concrete query formulation strategy provided from human experts, e.g., hypothetical document generation for `Contriever` and keyword-based rewriting for `bm25`. The rewriter is instructed to apply only this retriever-specific strategy.

Figure 4 illustrates the learning dynamics for `Contriever`. The general prompt achieves a reasonable initial performance but improves slowly, as the large space of rewriting choices makes exploration inefficient. The exploratory prompt yields the strongest initial performance and rapid improvement within the first 50 training steps. However, we observe that the model quickly converges to a suboptimal policy that emphasizes keyword matching, which it prematurely identifies as effective for `Contriever`. This early policy collapse leads to convergence around the 200th step. In contrast, the specific prompt, which enforces hypothetical document generation, achieves the best final performance. Although this setting starts from a relatively weaker initial point—due to the rewriter’s random guesses about the structure of target documents in the black-box corpus—it benefits from consistent guidance and gradually refines its strategy through GRPO rollouts, ultimately exhibiting

higher performance and slower convergence.

The conclusions differ for `bm25`, where the exploratory prompt outperforms the specific prompt, with both substantially outperforming the general prompt. Notably, even under exploratory prompts, the rewriter infers early in training that keyword-style queries are optimal for `bm25`. This can be attributed to `bm25` being a well-established retrieval algorithm whose underlying principles are extensively discussed in the pretraining corpora of LLMs. In contrast, general prompts yield the poorest initial performance, as `bm25`’s sensitivity to lexical choices causes many suboptimal rewrites to largely degrade early-stage rewards.

Taken together, these seemingly contrasting results reveal an intuitive principle: *when an LLM lacks sufficient prior knowledge or reasoning capability about a retriever’s optimal query strategy, explicit human expert guidance can effectively steer RL exploration and improve learning outcomes. Conversely, when the retriever’s behavior is already well captured in the model’s pretraining knowledge, exploratory prompts are sufficient—and can even outperform rigid, human-prescribed strategies.*

#### 4.4 Scaling of Model Size and Retrieval Steps

**RQ3.1: Can we enhance query formulation performance by scaling up model size?** We examine the effect of LLM model size on query formulation performance. As shown in Figure 4, increasing the rewriter size to 14B enables the model to break through the performance plateaus observed for the 4B and 8B variants. This behavior highlights the role of increased model capacity and world knowledge in enabling more diverse and effective exploration, thereby mitigating premature policy collapse during RL training.

Concretely, while prior knowledge—either learned implicitly (e.g., the 8B model discovering that longer queries are beneficial under general prompts) or provided explicitly via human expert guidance (e.g., document-style queries for `Contriever`)—suggests that longer or document-like queries are optimal, the 14B model goes beyond these assumptions. Under exploratory prompts, it discovers a distinct strategy that can be characterized as *statement-style* queries, which differ from both question-form and document-style rewrites. For example, given the original query *Is it possible to be white and Latino?*, a question-style rewrite such as *Can someone be both white and*

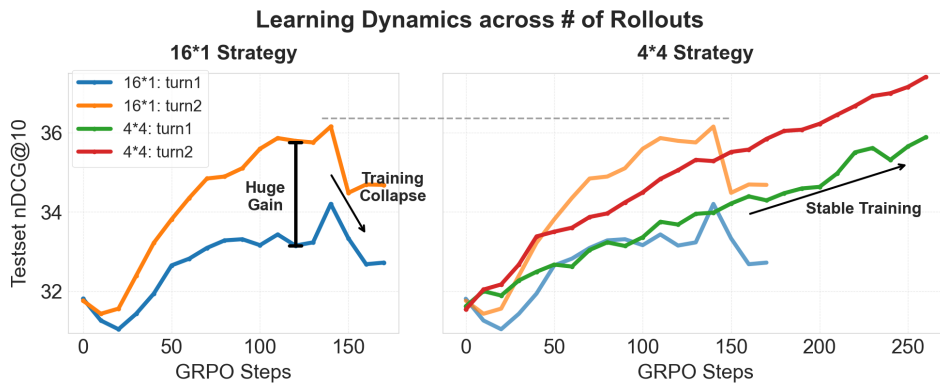


Figure 5: Training dynamics comparison of independent and branching rollout techniques

*Latino?*” achieves an nDCG@10 of 0.542, whereas a concise statement-style rewrite—*“Being white and Latino”*—surprisingly attains an nDCG@10 of 0.965. This strategy emerges naturally from RL training with a 14B model under exploratory prompts, as illustrated in Figure 2.

We further observe that the diversity of strategy exploration scales with model size. At the early stages of training, larger models exhibit lower initial average rewards, reflecting broader exploration over a wider range of candidate rewriting strategies, many of which are initially suboptimal. This increased exploratory behavior, however, ultimately enables the discovery of more effective query formulation policies.

Overall, the model scaling results reinforce insights from the prompt ablation studies in the previous section. When base models lack sufficient internal knowledge about retriever-specific behaviors, human expert-driven prompts can effectively constrain the search space and guide learning toward strong policies. In contrast, *larger and more capable models are able to discover non-obvious strategies that even human experts may not anticipate, allowing them to escape the local policy optima that constrain smaller models such as the 4B and 8B variants.*

**RQ3.2: Can we enhance query formulation performance by scaling up retrieval steps?** We observe that the  $4 \times 4$  Strategy (branching) significantly outperforms the  $16 \times 1$  Strategy (independent) in training stability. This performance gap stems from how the branching structure solves the credit assignment problem inherent in our shared reward function ( $R_{\text{total}} = 0.5R_{\text{turn1}} + R_{\text{turn2}}$ ).

The branching strategy isolates the learning signal for the second turn. By generating  $k = 4$  con-

tinuations from a fixed first-round outcome, the first-round reward becomes a constant  $C$  within that subgroup. When calculating the advantage for the second turn, this constant cancels out:

$$A_{\text{turn2}}^{(i)} \approx (0.5C + R_{\text{turn2}}^{(i)}) - \mathbb{E}_{\text{siblings}}[0.5C + R_{\text{turn2}}]. \quad (5)$$

This effectively removes the noise from the first turn, allowing the model to receive a precise gradient based solely on the quality of the second turn.

Simultaneously, the  $4 \times 4$  strategy stabilizes the first turn by acting as a Monte Carlo estimator for its value. In the  $16 \times 1$  strategy, the “future value” of a first-round rewrite is based on a single, noisy second-round sample. In contrast, the  $4 \times 4$  strategy estimates the value of a first-round action by adding its immediate reward to the average reward of the subsequent four second-round attempts that follow it. This tells the model: *“Given this specific first-round attempt, here is the expected outcome if we try 4 different second-round attempts.”* This averaging reduces variance in the reward signal for the first turn, preventing the training collapse observed in the independent baseline.

## 5 Related Work

Retrieval-augmented generation (RAG) is a standard approach for augmenting LLMs on tasks that exceed LLMs’ knowledge boundaries (Lewis et al., 2020; Xiao et al., 2025a). For bridging the behavioral gaps between LLMs and retrievers, prior approaches mitigate semantic mismatches via query rewriting (Ma et al., 2023; Chan et al., 2024) or zero-shot expansion (Gao et al., 2023). While Reinforcement Learning has been used to align rewrites with generation goals (Nogueira and Cho, 2017; Ma et al., 2023; Jin et al., 2025), these methods typically assume a uniform retrieval environment.

On the information retrieval side, prior works show that different retrievers display distinct length preferences (Thakur et al., 2021; Xiao et al., 2023). In this work, we systematically analyze *retriever-specific* query adaptation, addressing the “structural drift” where policies tailored for one backend fail to transfer to others.

Agentic frameworks like Self-RAG (Asai et al., 2024) and Adaptive-RAG (Mallen et al., 2023) utilize dynamic, multi-turn retrieval but often face training instability in long-horizon trajectories (Shao et al., 2023). Complementing research on the retriever-generator preference gap (Ke et al., 2024), we introduce a branching rollout technique that effectively stabilizes the optimization of multi-step query formulation strategies.

## 6 Conclusion

We present the first systematic study on adapting LLM query formulation to specific retrievers via Reinforcement Learning. We demonstrate that optimal strategies are highly retriever-dependent and quantify this stylistic transfer failure as “Structural Drift.” Furthermore, we find that scaling model size enables the discovery of non-intuitive strategies that outperform expert heuristics. Finally, our proposed branching rollout technique stabilizes multi-turn training, paving the way for robust, truly retriever-aware RAG systems.

## Limitations

First, our study focuses exclusively on text-only retrievers and text-based document collections, and does not consider multimodal retrievers (Xiao et al., 2025b; Chia et al., 2025; Xiao et al., 2025c) that can retrieve heterogeneous content such as images or audio. Extending our analysis to multimodal retrieval settings is an important direction for future work. Second, in our experiments on scaling the number of retrieval steps, we limit our evaluation to one-step and two-step retrieval. While these settings are sufficient to demonstrate the benefits of multi-step retrieval, it would be valuable to investigate learning dynamics and performance trends under longer retrieval trajectories in future studies.

## References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-rag: Learning to retrieve, generate, and critique through self-reflection.

In *The Twelfth International Conference on Learning Representations*.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, and 1 others. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

Vittorio Castelli, Rishav Chakravarti, Saswati Dana, Anthony Ferritto, Radu Florian, Martin Franz, Dinesh Garg, Dinesh Khandelwal, J Scott McCarley, Michael McCawley, and 1 others. 2020. The techqa dataset. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 1269–1278.

Chi-Min Chan, Chunpu Xu, Ruibin Yuan, Hongyin Luo, Wei Xue, Yike Guo, and Jie Fu. 2024. Rq-rag: Learning to refine queries for retrieval augmented generation. *arXiv preprint arXiv:2404.00610*.

Zhiyu Chen, Wenhua Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan R Routledge, and 1 others. 2021. Finqa: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3697–3711.

Yew Ken Chia, Liying Cheng, Hou Pong Chan, Maojia Song, Chaoqun Liu, Mahani Aljunied, Soujanya Poria, and Lidong Bing. 2025. M-longdoc: A benchmark for multimodal super-long document understanding and a retrieval-aware tuning framework. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 9244–9261.

Chanyeol Choi, Jihoon Kwon, Alejandro Lopez-Lira, Chaewoon Kim, Minjae Kim, Juneha Hwang, Jaeseon Ha, Hojun Choi, Suyeol Yun, Yongjin Kim, and 1 others. 2025. Finagentbench: A benchmark dataset for agentic retrieval in financial question answering. In *Proceedings of the 6th ACM International Conference on AI in Finance*, pages 632–637.

Robert Friel, Masha Belyi, and Atindriyo Sanyal. 2024. Ragbench: Explainable benchmark for retrieval-augmented generation systems. *arXiv preprint arXiv:2407.11005*.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. 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)*, pages 1762–1777.

Dan Hendrycks, Collin Burns, Anya Chen, and Spencer Ball. 2021. CUAD: An expert-annotated NLP dataset for legal contract review. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

- Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. [Unsupervised dense information retrieval with contrastive learning](#). *Trans. Mach. Learn. Res.*, 2022.
- Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*.
- Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 2567–2577.
- Ehsan Kamaloo, Aref Jafari, Xinyu Zhang, Nandan Thakur, and Jimmy Lin. 2023. Hagrid: A human-llm collaborative dataset for generative information-seeking with attribution. *arXiv preprint arXiv:2307.16883*.
- Zixuan Ke, Weize Kong, Cheng Li, Mingyang Zhang, Qiaozhu Mei, and Michael Bendersky. 2024. Bridging the preference gap between retrievers and llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10438–10451.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.
- Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting in retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5303–5315.
- Chaitanya Malaviya, Subin Lee, Sihao Chen, Elizabeth Sieber, Mark Yatskar, and Dan Roth. 2024. Expertqa: Expert-curated questions and attributed answers. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3025–3045.
- Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 9802–9822.
- Timo Möller, Anthony Reina, Raghavan Jayakumar, and Malte Pietsch. 2020. Covid-qa: A question answering dataset for covid-19. In *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*.
- Abhilash Nandy, Soumya Sharma, Shubham Madhaskhiya, Kapil Sachdeva, Pawan Goyal, and Niloy Ganguly. 2021. [Question answering over electronic devices: A new benchmark dataset and a multi-task learning based QA framework](#). In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4600–4609, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Rodrigo Nogueira and Kyunghyun Cho. 2017. Task-oriented query reformulation with reinforcement learning. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pages 574–583.
- Stephen E. Robertson and Hugo Zaragoza. 2009. [The probabilistic relevance framework: BM25 and beyond](#). *Found. Trends Inf. Retr.*, 3(4):333–389.
- Sentence-Transformers. 2025. all-minilm-l6-v2: Sentence embeddings model. <https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2>. Accessed: 2026-01-06.
- Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9248–9274.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. [Deepseekmath: Pushing the limits of mathematical reasoning in open language models](#). *CoRR*, abs/2402.03300.
- Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*.
- Chenghao Xiao, Hou Pong Chan, Hao Zhang, Mahani Aljunied, Lidong Bing, Noura Al Moubayed, and Yu Rong. 2025a. Analyzing llms’ knowledge boundary cognition across languages through the lens of internal representations. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24099–24115.
- Chenghao Xiao, Hou Pong Chan, Hao Zhang, Weiwen Xu, Mahani Aljunied, and Yu Rong. 2025b. Scaling language-centric omnimodal representation learning. *arXiv preprint arXiv:2510.11693*.
- Chenghao Xiao, Isaac Chung, Imene Kerboua, Jamie Stirling, Xin Zhang, Márton Kardos, Roman Soltanin, Noura Al Moubayed, Kenneth Enevoldsen, and Niklas Muennighoff. 2025c. Mieb: Massive image embedding benchmark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22187–22198.

Chenghao Xiao, Yizhi Li, G Hudson, Chenghua Lin, and Noura Al Moubayed. 2023. Length is a curse and a blessing for document-level semantics. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1385–1396.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2369–2380.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, and 1 others. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021. Tat-qa: A question answering benchmark on a hybrid of tabular and textual content in finance. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3277–3287.

## A RAGBench Details

RAGBench (Friel et al., 2024) is a large-scale RAG benchmark dataset which comprises more than 100,000 samples across diverse domains such as finance, law, and healthcare. Specifically, RAGBench collects extensive QA datasets, including CovidQA (Möller et al., 2020), PubMedQA (Jin et al., 2019), HotpotQA (Yang et al., 2018), MS Marco (Bajaj et al., 2016), CUAD (Hendrycks et al., 2021), EManual (Nandy et al., 2021), TechQA (Castelli et al., 2020), FinQA (Chen et al., 2021), TAT-QA (Zhu et al., 2021), ExpertQA (Malaviya et al., 2024), and HAGRID (Kamalloo et al., 2023).

## B FinAgentBench and its Adaptation

FinAgentBench (Choi et al., 2025) is a financial information retrieval benchmark which contains over 18,000 samples curated by experts from SEC filings between 2023 and 2024. In the original benchmark, each sample is formulated as a single large prompt that combines ranking instructions, a financial question, and a set of candidate document chunks, which are annotated with graded relevance labels. To adapt FinAgentBench to a standard retrieval setting, we decompose each prompt into its constituent components and extract unique questions and document chunks across all samples. Because the same question may appear in multiple samples (e.g., when posed over different filings), and the same set of document chunks may be reused across samples (e.g., when multiple questions are asked about the same filing), we pool questions and chunks globally and retain only unique instances via exact matching, without any additional normalization or model-based processing. We then inherit relevance annotations from the original benchmark by mapping each question-chunk pair to its corresponding relevance label, thereby constructing a conventional retrieval dataset consisting of questions, documents, and graded relevance judgments.

## C RE-MMD Empirical Estimation

In Section 4.2, we defined RE-MMD as the distance between mean embeddings in a high-dimensional feature space. Directly computing the embedding  $\phi(q)$  is often intractable. Therefore, we apply the kernel trick  $k(q, q') = \langle \phi(q), \phi(q') \rangle_{\mathcal{H}}$  to estimate the distance empirically.

Method	Retrieval (nDCG at 10)	Accuracy	F1	EM
<b>Full RAG (w/ top 20 docs)</b>				
no RAG	-	25.3	26.0	18.8
+ vanilla RAG	41.01	41.4	41.0	31.2
+ general rewrite w/ Gemini	39.16	42.3	42.0	32.0
+ <b>(ours)</b> retriever-specific rewrite	<b>52.44</b>	<b>46.9</b>	<b>46.0</b>	<b>35.2</b>
<b>Full RAG (w/ top 5 docs)</b>				
no RAG	-	25.3	26.0	18.8
+ vanilla RAG	41.01	36.5	36.0	27.3
+ general rewrite w/ Gemini	39.16	38.1	38.0	28.5
+ <b>(ours)</b> retriever-specific rewrite	<b>52.44</b>	<b>43.7</b>	<b>43.0</b>	<b>32.8</b>

Table 4: Performance comparison of Full RAG setups using top 20 and top 5 retrieved documents.

Given a set of successful query trajectories  $Q_{src} = \{x_1, \dots, x_n\}$  from  $\mathcal{E}_{src}$  and  $Q_{tgt} = \{y_1, \dots, y_m\}$  from  $\mathcal{E}_{tgt}$ , we employ the unbiased estimator:

$$\begin{aligned}
RE-MMD^2 = & \frac{1}{n(n-1)} \sum_{i \neq j} k(x_i, x_j) \\
& + \frac{1}{m(m-1)} \sum_{i \neq j} k(y_i, y_j) \quad (6) \\
& - \frac{2}{nm} \sum_{i,j} k(x_i, y_j)
\end{aligned}$$

To diagnose the nature of the drift, we utilize two distinct kernel functions:

**1. Semantic Kernel.** To measure intent drift, we use the Gaussian RBF kernel over dense embeddings (e.g., MiniLM):

$$k_{sem}(x, y) = \exp\left(-\frac{\|\phi_{sem}(x) - \phi_{sem}(y)\|^2}{2\sigma^2}\right) \quad (7)$$

This captures non-linear semantic similarities between queries.

**2. Structural Kernel.** To measure lexical/stylistic drift, we use a Linear kernel over sparse lexical features (TF-IDF):

$$k_{struct}(x, y) = \langle \phi_{struct}(x), \phi_{struct}(y) \rangle \quad (8)$$

This is equivalent to the Euclidean distance of centroids in the lexical space, capturing explicit divergence in vocabulary usage and query formulation style.

## D Full RAG Performance

To validate whether the performance gains observed in retriever-specific retrieval translate into improvements in the full-RAG pipeline, we conduct a series of full-RAG experiments using HotpotQA as an example. Specifically, we adopt Contriever as the retriever and Qwen2.5-7B-Instruct as the generator, and compare our rewriter agent against vanilla RAG and general rewrites (non-retriever-specific) from Gemini-2.5-flash. We use the HotpotQA corpus from BEIR as the external knowledge base and the original HotpotQA questions as queries. Following standard RAG evaluation protocols, we report Accuracy, F1, and Exact Match (EM).

As shown in Table 4, enhancements in nDCG@10 consistently translates to performance gains in the full RAG performance. Notably, these improvements remain stable across different numbers of retrieved documents, indicating the robustness of retriever-aware query rewriting.

## E Zero-shot Multi-turn Retrieval Case Study

We provide an illustrative example of multi-turn retrieval using our rewrite agent over four retrieval turns. Although the model is trained using only a two-turn branching strategy in our experiments, it demonstrates the ability to generalize to longer multi-turn interactions in a zero-shot manner. This suggests that the learned retrieval behaviors are not limited to the training horizon and can effectively extend to more complex interaction settings.

**Original Query:** "What would be the effects on the us dollar if hypothetically it is no longer regarded as the world reserve currency?"

**Rewrite Turn 1 (Hypothetical Document Generation):**

The US dollar's status as the world's primary reserve currency has been a cornerstone of global economic stability for decades. If the dollar were hypothetically no longer regarded as the world reserve currency, the implications would be profound and multifaceted. First, the international demand for the dollar would likely decline, leading to a depreciation in its value. This depreciation could increase the cost of imports for the United States, contributing to inflation. Conversely, US exports might become more competitive, potentially boosting the economy in the short term.

Central banks and governments around the world currently hold significant reserves of US dollars, using them for trade settlements, investments, and as a safe asset. A shift away from the dollar would require these entities to diversify their reserves, possibly increasing the role of other currencies such as the euro, the Chinese yuan, or the Japanese yen. This transition could lead to the emergence of new global financial centers and alter the balance of economic power.

The United States would also face challenges in maintaining its influence over global financial institutions, such as the International Monetary Fund (IMF) and the World Bank, which have historically been dominated by US interests. A reduced role of the dollar could diminish the US's ability to shape international economic policy, affecting everything from trade agreements to financial regulations.

Moreover, the US government's capacity to borrow at low interest rates might be impacted, as the dollar's status as a safe haven could diminish. This could lead to higher borrowing costs for the US, affecting both the federal budget and private sector investments. In the long term, the global economy might become more fragmented, with regional currencies playing a larger role in international trade, potentially leading to increased economic volatility.

While the transition away from the dollar as the reserve currency would present challenges, it could also offer opportunities for the US to innovate in financial technology and strengthen its economic partnerships with other nations. The ultimate impact would depend on how smoothly the transition is managed and the policies implemented to support a new global financial order.

**Analysis:** The rewriter uses the **hypothetical document generation** technique in turn 1, but the terms are too general, yielding a **nDCG@10 of 0.0** in the first round. However, the rewriter is able to see the returned documents and learns how documents in the corpus look like.

**Rewrite Turn 2 (Entity Anchoring):**

What are the economic implications of **de-dollarization** and the rise of alternative reserve currencies, such as the **Chinese yuan**, on **global financial systems** and the United States?

**Analysis:** Feedback from turn 1, although low scoring, exposes specific entities in the returned documents (omitted here). Thus, in the second round, the rewriter pivots to more specific directions, such as "de-dollarization", "Chinese yuan", and "global financial systems", enhancing the **nDCG@10 to**

56.84.

**Rewrite Turn 3 (Further Enhancement):**

{omitted for brevity}

**Rewrite Turn 4 (Domain/Style Alignment):**

What are the financial and economic ramifications for the United States and global markets if the US dollar were no longer the dominant reserve currency, including impacts on foreign exchange reserves, **corporate financial strategies**, and the role of alternative currencies like the Chinese yuan?

**Analysis:** After understanding the genre of available documents (corporate 10-K filings and financial disclosures) in this corpus, the agent adopts specific financial reporting terminology and be more specific about the domain, adapting to the environment style. The final rewrite enhances the **nDCG@10 to 93.74**, a near perfect score.

## F Training Prompts

Figure 6 describes the training prompts used in different settings of our experiments.

**General Prompt:**

In order to retrieve supporting documents to answer the given question, you need to rewrite and extend a given question into one new query. Remember that the rewritten query is not limited to a simple question, but can be a passage including multiple sentences that provides context for the input question. Here are some commonly used query rewriting methods, please select one appropriate method and use it to generate one new rewritten query:

1. Query Understanding: Extract core information from the query, and understand the core elements. For example, 'What was the federal alternative minimum tax credit carryforwards?' can be phrased as 'Federal, U.S. federal tax system; Alternative Minimum Tax (AMT), a parallel tax system designed to ensure taxpayers pay a minimum level of tax; Credit carryforwards, unused credits that can be carried forward to future years to offset regular tax liability;'.
2. Semantic Rewriting: (1) Identify important entities and expand with related information. For example, 'Steve Jobs' to 'Apple CEO Steve Jobs'. (2) Add broader (hypernym) or more specific (hyponym) concepts. For example, 'husky' to 'pet dog husky'. (3) Change word order or phrasing to match common expressions in target documents. For example, 'make steak' to 'steak recipe tutorial'.
3. Query Decomposition: Break down a complex query into multiple sub-queries for separate retrieval. For example, 'Which stock price is higher, Apple or Google?' to '<rewrite>what is the stock price of Apple?</rewrite><rewrite>what is the stock price of Google?</rewrite>'. Also consider more complex queries that can benefit from breaking down into multi-hop queries.
4. General Rewriting: (1) Paraphrasing: rephrase the whole query if you think can possibly help the embedding model generate more accurate embeddings. For example, 'How to lose weight effectively?' to 'What are some scientifically proven weight loss methods?'. (2) Intent Refinement, infer the user true intent and rewrite the query accordingly. (3) Multi-hop Query Generation, generate indirectly related queries that help improve recall. For example, 'diabetic diet' to 'fruits suitable for diabetics'.
5. Hypothetical Document Generation: Generate a hypothetical target document, in order to use this hypothetical document to retrieve target document. For example, (1) Given a question asking about the law clause related to a case, directly generate a hypothetical clause for retrieval. (2) Given a question about a company's stock, directly generate a hypothetical stock price report that looks alike the target document that contains the information.

Before outputting the rewritten query, you need to think carefully about choosing which rewriting method, why choose this method, and how to use it. And please enclose your think process within <think> and </think>. Then the rewritten query must be enclosed between <rewrite> and </rewrite>. Do not put the thinking process between <rewrite> and </rewrite>. For methods other than Query Decomposition, only one rewrite query is needed. If the question is rewritten as multiple queries using one rewriting method (for example, when using Query Decomposition), please enclose each different query with separate <rewrite> and </rewrite>. Output can look like: <think> your think process </think><rewrite> rewritten query </rewrite>. User Question:

**Exploratory Prompt (Contriever):**

You are using a retriever model to retrieve relevant documents to answer a question. The retriever model you will be using is Contriever, which is trained using two spans of a Wikipedia article as positive pairs. You will need to think about the best search strategy using Contriever and rewrite the original query into a query that you think yields the best retrieval results for Contriever. Before outputting the rewritten query, you need to think carefully about what to write to look like target documents, enclose your think process within <think> and </think>. Then the rewritten query must be enclosed between <rewrite> and </rewrite>. Do not put the thinking process between <rewrite> and </rewrite>. Output can look like: <think> your think process </think><rewrite> rewritten query </rewrite>. User Question:

**Exploratory Prompt (bm25)**

You are using a retriever system to retrieve relevant documents to answer a question. The retriever system you will be using is BM25, which is a sparse retrieval system. You will need to think about the best search strategy using BM25 (e.g., keyword-based) and rewrite the original query into a query that you think yields the best retrieval results for BM25. Before outputting the rewritten query, you need to think carefully about what to write to yield the best retrieval results, enclose your think process within <think> and </think>. Then the rewritten query must be enclosed between <rewrite> and </rewrite>. Do not put the thinking process between <rewrite> and </rewrite>. Output can look like: <think> your think process </think><rewrite> rewritten query </rewrite>. User Question:

**Direct/Specific Prompt (Contriever)**

You are using a retriever model to retrieve relevant documents to answer a question. The retriever model you will be using is Contriever, which is trained on using two spans of a Wikipedia article as positive pairs. Therefore, it works best for using document to retrieve document. You will be given a user question, try using the "Hypothetical Document Generation" technique to generate a hypothetical target document, which will be used to retrieve target similar documents. For example, (1) Given a question asking about the law clause related to a case, directly generate a hypothetical clause for retrieval. (2) Given a question about a company's stock, directly generate a hypothetical stock price report that looks alike the target document that contains the information. Before outputting the hypothetical document, you need to think carefully about what to write to look like target documents, enclose your think process within <think> and </think>. Then the hypothetical document must be enclosed between <rewrite> and </rewrite>. Do not put the thinking process between <rewrite> and </rewrite>. Output can look like: <think> your think process </think><rewrite> hypothetical document </rewrite>. User Question:

**Direct/Specific Prompt (bm25)**

You are a query optimization assistant for a BM25 retrieval system. Since BM25 is a sparse, keyword-based model, it performs best with precise search terms rather than natural language. Your task is to rewrite the user's original query into a keyword-optimized version that maximizes retrieval performance. You must specifically optimize for term morphology and spelling variations (such as singular vs. plural forms or compound word handling). Because BM25 relies on exact token matching to calculate Inverse Document Frequency (IDF), using the most standard or canonical form of a word is critical for aligning with the target corpus. Output the best possible search query for this system. Before outputting the rewritten query, you need to think carefully about what to write to yield the best retrieval results, enclose your think process within <think> and </think>. Then the rewritten query must be enclosed between <rewrite> and </rewrite>. Do not put the thinking process between <rewrite> and </rewrite>. Output can look like: <think> your think process </think><rewrite> rewritten query </rewrite>. User Question:

Figure 6: Training prompts used in our different settings.