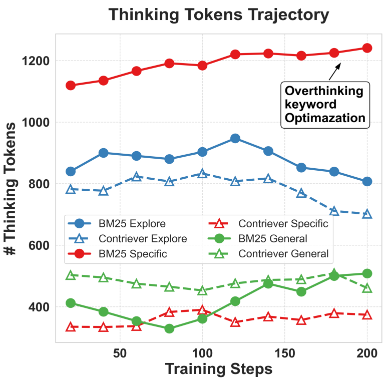
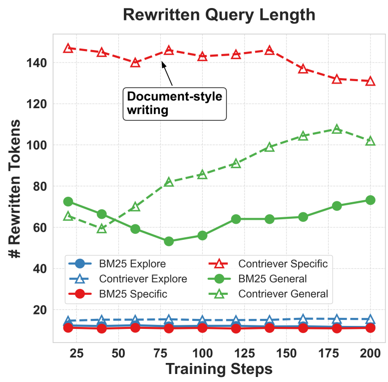
Understanding the Behaviors of Environment-aware Information Retrieval

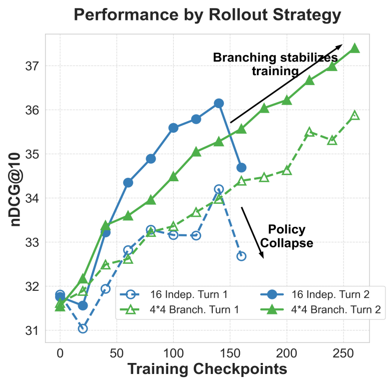
Anonymous ACL submission



(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 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.

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 π_θ 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 $\{q_1, ..., q_G\}$ from the policy $\pi_{\theta_{old}}$. The GRPO objective is maximizing:

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

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:

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

where μ_{group} and σ_{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 $\tau_i = [q_{i,1}, e_{i,1}, q_{i,2}, e_{i,2}, ..., q_{i,K}]$, 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:

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

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., 16×1), we implement a branching structure (e.g., 4×4) 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:

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

- **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

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.

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-sround 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` (Team of Sentence-Transformers, 2025), `Contriever` (Izacard et al., 2022), and `BM25` (Crestani et al., 1998).

`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 using 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 systmes, such as using mean pooling for Contriever, and using the query instruction template for Qwen3-Embedding, etc.

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

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	**72.1**	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	**64.0**	22.9	**87.9**	**68.1**	76.0	69.9	13.9	30.9	55.2↑
Contriever + rewrite direct + turn2	**42.6**	70.1	**71.5**	63.5	**23.3**	86.1	64.9	**78.5**	**73.0**	**15.0**	**32.5**	**56.5↑**
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	**40.5**	**66.1**	**66.3**	**46.1**	**33.4**	**77.9**	**82.6**	**72.5**	**61.9**	**24.9**	**53.7**	**56.9↑**
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	**41.4**	**72.1**	**71.1**	**61.2**	**13.5**	**86.4**	**61.1**	**81.8**	**75.4**	**13.7**	**31.4**	**55.4↑**
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	**60.5**	**83.4**	**77.4**	**66.9**	**30.0**	**89.4**	**68.0**	**84.5**	**83.2**	**20.1**	**41.2**	**64.1↑**

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	**49.23↑**	**32.08↑**	26.71	**36.28↑**	**52.44↑**	**16.17↑**	**7.68↑**	77.27	**32.18↑**	**13.10↑**	**37.52↑**	**11.75↑**	**64.54↑**	**34.98↑**
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	**7.39↑**
BM25	8.17
BM25 + rewrite	**9.02↑**
all-MiniLM-L6-v2	6.69
all-MiniLM-L6-v2 + rewrite	**7.17↑**
Qwen3-Embedding-0.6B	9.10
Qwen3-Embedding-0.6B + rewrite	**10.81↑**

Table 3: nDCG@10 performance on FinAgentBench.

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.

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 ϕ_{sem} (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 ϕ_{struct}.

Formally, we define the squared RE-MMD distance in a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$ as: $\text{RE-MMD}^2(\mathcal{E}_{\text{src}}, \mathcal{E}_{\text{tgt}}) = \left\|\mu_{\text{src}} - \mu_{\text{tgt}}\right\|_{\mathcal{H}}^2$, 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.

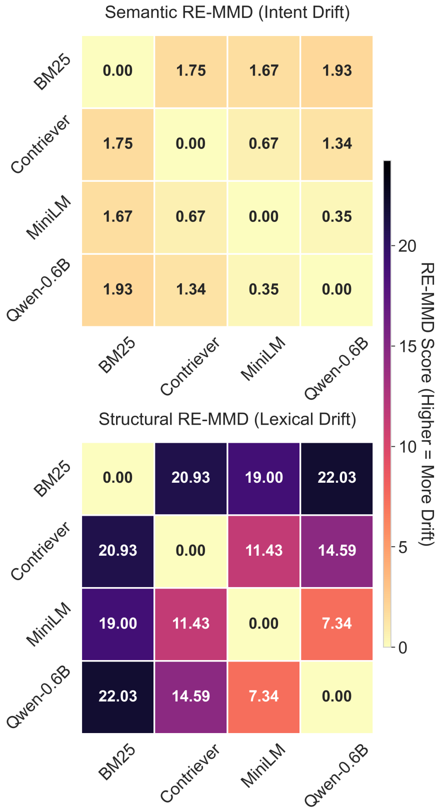


Figure 3: Measuring semantic and structural drifts of optimal query across retrieval environments. Scores are our proposed RE-MMD multiplied by 100.

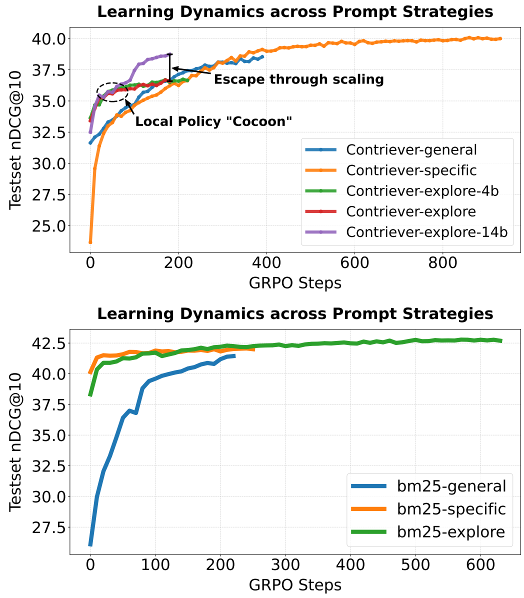


Figure 4: Learning dynamics across different prompt methods and model sizes of query rewriter LLM.

Results. Figure 5 reports the RE-MMD scores (times 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 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 comprehensiveness:

- **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 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

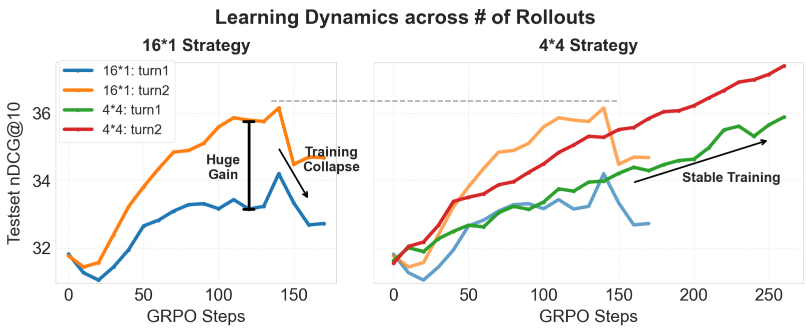


Figure 5: Training dynamics comparison of independent and branching rollout techniques

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 4×4 Strategy (branching) significantly outperforms the 16×1 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 $k = 4$ continuations 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:

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

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 4×4 strategy stabilizes the first turn by acting as a Monte Carlo estimator for its value. In the 16×1 strategy, the "future value" of a first-round rewrite is based on a single, noisy second-round sample. In contrast, the 4×4 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

Prior approaches mitigate semantic mismatches via pipeline rewriting (Ma et al., 2023) or zero-shot expansion (Gao et al., 2023). While Reinforcement Learning has been used to align rewrites with generation goals (Nogueira and Cho, 2017; Chan et al., 2024), these methods typically assume a uniform retrieval environment. In contrast, we systematically analyze *retriever-specific* 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 (Xu 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 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, and 1 others. 2024. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *ICLR*.

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, Weize Wang, Hao Cheng, and 1 others. 2024. Rq-rag: Learning to refine queries for retrieval augmented generation. *arXiv preprint arXiv:2404.00610*.

Zhiyu Chen, Wenhu 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.

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.

Fabio Crestani, Mounia Lalmas, C. J. van Rijsbergen, and Iain Campbell. 1998. "is this document relevant? ... probably": A survey of probabilistic models in information retrieval. *ACM Comput. Surv.*, 30(4):528–552.

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 *ACL*.

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.

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 Kamalloo, 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*.

Xinbei Ma, Yeyun Gong, Pengcheng He, and 1 others. 2023. Query rewriting for retrieval-augmented large language models. *EMNLP*.

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, and 1 others. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *ACL*.

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 Maddhashiya, 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 *EMNLP*.

Zhihong Shao, Yeyun Gong, Yelong Shen, and 1 others. 2023. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. *Findings of EMNLP*.

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.

Team of Sentence-Transformers. 2025. all-minilm-l6-v2: Sentence embeddings model. `https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2`. Accessed: 2026-01-06; pretrained sentence-transformers model mapping sentences & paragraphs to 384-dimensional vectors for semantic tasks.

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*.

Zhaozhuo Xu, Zirui Liu, Beidi Grundtt, and 1 others. 2024. Bridging the preference gap between retrievers and llms. *ACL*.

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 Appendix: RAGBench

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 Appendix: 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 questions, documents, and graded relevance judgments.

C Appendix: 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.

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

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

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):

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

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):

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

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