MoR: Better Handling Diverse Queries with a Mixture of Sparse, Dense, and Human Retrievers

Jushaan Kalra[†1*] **Xinran Zhao**[†1] **To Eun Kim**[1] **Fengyu Cai**[2]
Fernando Diaz[1] **Tongshuang Wu**[1]
[1]Carnegie Mellon University, [2]Technical University of Darmstadt

Abstract

Retrieval-augmented Generation (RAG) is powerful, but its effectiveness hinges on which retrievers we use and how. Different retrievers offer distinct, often complementary signals: BM25 captures lexical matches; dense retrievers, semantic similarity. Yet in practice, we typically fix a single retriever based on heuristics, which fails to generalize across diverse information needs. Can we dynamically select and integrate multiple retrievers for each individual query, without the need for manual selection? In our work, we validate this intuition with quantitative analysis and introduce a *mixture of retrievers*: a zero-shot, weighted combination of heterogeneous retrievers. Extensive experiments show that such mixtures are effective and efficient: Despite totaling just 0.8B parameters, this mixture outperforms every individual retriever and even larger 7B models—by +10.8% and +3.9% on average, respectively. Further analysis also shows that this mixture framework can help incorporate specialized non-oracle *human* information sources as retrievers to achieve good collaboration, with a 58.9% relative performance improvement over simulated humans alone.

1 Introduction

Although Retrieval Augmented Generation (RAG) (Lewis et al., 2020) has been shown to improve the reliability and reduce the hallucination of large language models (LLMs), no single retriever is optimal for all queries. For example, in encyclopedic question answering tasks such as Natural Questions (Kwiatkowski et al., 2019), embedding-based retrievers like DPR (Karpukhin et al., 2020b) often outperform token-based approaches like BM25 (Robertson and Zaragoza, 2009). In contrast, in specialized domains such as medicine and biology, token-based approaches remain a strong baselines (Thakur et al., 2021). Because retrieval effectiveness can vary significantly across domains and even across queries, finding a universally optimal retriever remains an open problem, highlighting the importance of understanding query-level comparative advantages among retrievers, rather than relying solely on aggregate performance. As a result, to accommodate real-world applications with diverse query types (Sawarkar et al., 2024), a core challenge is to select appropriate retrievers and combine their content.

In this work, we propose a solution in the form of **Mixture-of-Retrievers** (`MoR`) framework. Inspired by the Mixture-of-Experts architecture (Jacobs et al., 1991; Shazeer et al., 2017), `MoR` dynamically selects and combines retrievers for each query by leveraging signals collected both before and after retrieval, eliminating the need for manual retriever selection. Specifically, as shown in Figure 1, `MoR` adopts a multi-granularity retrieval strategy (Chen et al., 2023b) to expand the pool of retrievers and exploit their complementary strengths operating on different semantic units. Based on work from aggregated search (Arguello and Diaz, 2013), we consider retriever-trustworthiness signals at different stages: (i) pre-retrieval signals, where we extend the notion of model familiarity from the *when-to-retrieve* literature (Mallen et al., 2023; Zhao et al., 2023) to the retriever-level, and (ii) post-retrieval signals, where we design signals akin to query performance prediction (Diaz, 2007; Long and Chang, 2014; Roitman, 2017; Arabzadeh et al., 2024). Using these signals, we compute per-query, per-retriever weights subsequently used to adjust the relevance scores, enabling effective re-ranking of the aggregated retrieval results from the entire pool of retrievers.

We conduct extensive experiments on four complex scientific domain retrieval tasks featuring diverse queries (Boteva et al., 2016; Cohan et al.,

* † denotes equal contribution. Corresponding contact email addresses: {xinranz3,sherryw}@andrew.cmu.edu.
Our code is available at https://github.com/Josh1108/MixtureRetrievers.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 11971–11990
November 4-9, 2025

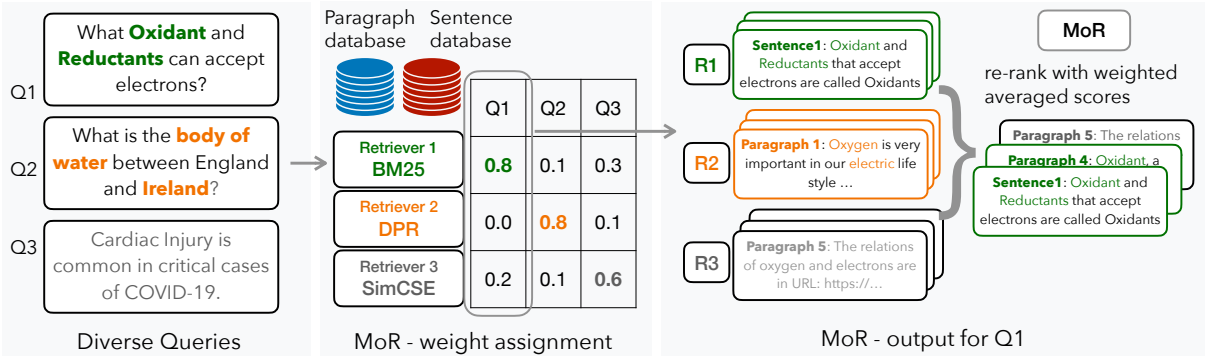

Figure 1: We demonstrate Mixture of Retrievers (MoR), which handles diverse queries—such as keyword-based, general, and factual questions—by combining multiple retrievers: BM25, DPR, and SimCSE. MoR assigns trustworthiness weights to each retriever in a zero-shot manner, then aggregates and re-ranks document scores across databases of varying granularity to produce the final output.

2020; Wadden et al., 2020; Welbl et al., 2017). Our results show that variants of MoR outperform individual retrievers, including the state-of-the-art GritLM (Muennighoff et al., 2024) in both retrieval and RAG tasks. This shows that MoR determines the strengths and weaknesses of different retrievers and effectively automates retriever selection.

Notably, our further analysis demonstrates that the mixture and fusion strategy generalizes across a wide range of retriever types, including token-based, embedding-based, and even "human retrievers", where humans provide noisy but useful information in response to queries. This highlights the potential of human-LLM collaboration in knowledge-intensive tasks, achieving on average 1.4× performance of MoR alone and a +58.9% relative gain over purely simulated human responses.

In summary, our main contributions are:

- We identify the query-level comparative advantages and propose the MoR architecture to improve retrieval performance with *folk wisdom*.
- We design a zero-shot method to construct MoR with multi-granularity deep fusion, pre-retrieval signals, and post-retrieval performance prediction, which achieves improved performance and robustness over existing retrievers.
- Further analysis on simulated noisy human experts shows MoR can potentially help estimate human retriever trustworthiness from a give corpus, and improve the human-LLM collaboration on knowledge-intensive tasks.

2 Related Work

Adaptive treatment in RAG. Recent advancements in RAG have increasingly adopted to apply diverse strategies to enhance adaptability across diverse tasks. Sawarkar et al. (2024) seek to combine lexical and semantic retrievers via designing fixed query types. In parallel, various studies have investigated adaptive query treatment mechanisms for differen purposes, including identifying relevant knowledge sources (Guerraoui et al., 2025; Lee et al., 2024), adapting retrieval strategies based on query complexity (Mallen et al., 2023; Zhao et al., 2023; Jeong et al., 2024), optimizing retrieval cost-effectiveness (Mu et al., 2024), supporting multimodal tasks (Yeo et al., 2025) and ranking retrievers through learning (Kim and Diaz, 2025). Instead of choosing fixed specific retrievers, our work dynamically utilizes the information from all retrievers to combine their complementary advantages.

Distributed Information Retrieval. Advances in the adaptive RAG systems are deeply rooted in foundational research in information retrieval (IR), particularly the field of distributed information retrieval. Concepts such as resource selection (Dai et al., 2017; Khramtsova et al., 2023), distributed and federated search (Callan et al., 1995; Callan, 2002; Diaz et al., 2010), meta-search (Glover et al., 1999; Chen et al., 2001), and query performance prediction (Diaz, 2007; Long and Chang, 2014; Roitman, 2017; Singh et al., 2023; Arabzadeh et al., 2024) for determining where to search are directly relevant to the modern multi-retriever set up in RAG systems. Similarly, ideas from aggregated search (Arguello et al., 2017), and rank fusion techniques (Cormack et al., 2009) have informed how retrieval signals should be combined across multiple sources, highlighting the deep connections

between routing in RAG architectures and traditional IR research. In our work, we study how the semantics of dense retriever representation space serve as a strong zero-shot signal to combine information from multiple sources.

3 Any folk wisdom among retrievers?

Following existing practice (Ngo et al., 2022; Wang et al., 2023), we first quantitatively validate our assumption that retrievers have varying strengths, and establish the potential of mixturing through simple query routing.

3.1 Preliminary

The task of retrieval. We consider a set of retrievers $L_{\mathcal{R}}$, where each retriever $\mathcal{R}_i \in L_{\mathcal{R}}$. Then the task of a single retriever is: given a query q and a corpus $\mathcal{D} = \{d_j\}$, each retriever $\mathcal{R}_i$ assigns a relevance score to each document d_j using its own scoring function $s_i(q, d_j)$ (e.g., cosine similarity). This results in a score vector $\mathbf{s}_i \in \mathbb{R}^{|\mathcal{D}|}$, where $|\mathcal{D}|$ is the size of the corpus. These scores can then be used to rank the documents.

Datasets and metrics. Then we build a test bed for various retrievers on retrieval tasks with diverse query types, i.e., compared to encyclopedic questions in Natural Questions (Kwiatkowski et al., 2019), these tasks involve complex query-document relations, e.g., there are multiple conditions organized in a first-order logic (Cai et al., 2024). Specifically, with similar setting as BEIR (Thakur et al., 2021), we consider: (i) NFCorpus (Boteva et al., 2016): a medical retrieval dataset with non technical natural language queries and a complex terminology-heavy corpus with medical documents; (ii) SciDocs (Cohan et al., 2020): a scientific document retrieval task with diverse subtasks such as citation prediction, paper recommendation, etc, with a corpus of documents from more than ten domains including art, business, computer science, geology, etc; (iii) SciFact (Wadden et al., 2020): a retrieval task with expert-written scientific claims and evidential abstracts as the corpus; and (iv) SciQ (Welbl et al., 2017): a large retrieval dataset with domain-specific text and science exam style questions. Further statistics and details are shown in Table 9 in the appendix. Following (Cai et al., 2024), we use Normalized Discounted Cumulative Gain (NDCG@K) to compare the retrieval performance, where K denotes the number of top retrieved documents considered.

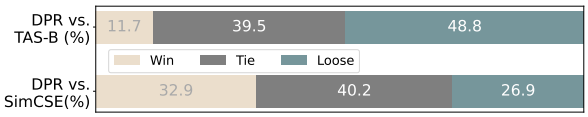

Figure 2: Performance comparison of different retrievers on SciFact. *A Wins B* denotes that the gold entry appears in the top 20 retrieved documents of A, not B. The rates denote the micro-average of all queries, which shows clear comparative advantages.

A diverse set of retrievers. We consider both sparse and various BERT-sized (Devlin et al., 2019) dense retrievers as candidates of $L_{\mathcal{R}}$ to create our mixture of retrievers, including (1) unsupervised: BM25 (Trotman et al., 2014), SimCSE (Gao et al., 2021), and Contriever (Izacard et al., 2022); (2) supervised: DPR (Karpukhin et al., 2020a), ANCE (Xiong et al., 2021), TAS-B (Hofstätter et al., 2021), GTR (Ni et al., 2022), and MPNet (Song et al., 2020). The cumulative number of parameters of $L_{\mathcal{R}}$ is 836 million, i.e., 0.836 B. In addition, we also set a competitive performance reference with various large-language-model-based retrievers with 7 billion (7B) parameters: RepLLaMA (Ma et al., 2023) and GritLM (Muennighoff et al., 2024).

As depicted in Table 1, the above retrievers constitute a diverse set that varies in parameter sizes, backbone architectures, and training signals. We include further detailed textual descriptions and model checkpoints we used in Appendix A.3.

3.2 Validating the folk wisdom

With the experiment suite in hand, we first validate our assumption of performance gain from the comparative advantages among retrievers with diverse queries in the same domain. We take SciFact (Wadden et al., 2020) as an example to compare the win rate of retriever pairs across all queries. As shown in Figure 2, despite the large gap in overall performance, DPR can still win on 11.7% queries compared to more advanced TAS-B (Hofstätter et al., 2021), suggesting a corresponding performance boost if we are able to select the combine the retrievers. Similarly, for retrievers with drastically different characteristics, e.g., supervised DPR and unsupervised SimCSE (Gao et al., 2021), despite similar overall performance, there are significant comparative advantages.

We further verify the *folk wisdom* by considering an analytical case of a mixture of retrievers with candidates $L_{\mathcal{R}}$, i.e., query routing. For each query, we consider a simplified case by selecting the best-

Retriever	Type	Backbone	Params	Training Signal	Strength
Retrievers relying on general-purpose embeddings					
BM25	Sparse	TF-IDF	N/A	None (rule-based)	Strong zero-shot; fast and interpretable
SimCSE	Dense	BERT-base	110M	Self-supervised contrastive on Wikipedia	Simple training; effective sentence encoder
Contriever	Dense	BERT-base	110M	Unsupervised contrastive (web + Wikipedia)	No labels needed; strong zero-shot
Retrievers trained on query-document pairs					
DPR	Dense	Dual BERT-base	110M	Supervised on QA (e.g., SQuAD)	Strong in-domain accuracy; widely adopted
ANCE	Dense	Dual BERT-base	110M	Contrastive with ANN negatives	Hard negative mining improves retrieval quality
TAS-B	Dense	BERT-base	66M	Distilled from ColBERT	Efficient; distilled from interaction-heavy model
GTR	Dense	T5-base	220M	QA pretrain + MS MARCO finetune	Generalizes well
MPNet	Dense	MPNet	110M	Permutation + position-aware pretraining	Strong encoder for semantic similarity
Retrievers based on large language models					
RepLLaMA	Dense	LLaMA-7B	7B	Supervised DPR-style dual-encoder	High quality; strong long-context handling
GritLM	Dense	7B LM	7B	Joint generative + embedding training	SOTA on MTEB (Muennighoff et al., 2022); dual-purpose model

Table 1: Comparison of retrievers used in or compared with our mixture. We include both sparse and dense models, varying in architecture, size, and training signals. Additional details are in Section A.3.

Retriever	NFCorpus	SciDocs	SciFact	SciQ
RepLlama-7b	38.1	18.1	76.0	65.9
GritLM-7b	38.8	27.7	79.8	79.7
Route Oracle	45.1	35.1	84.4	96.6

Table 2: NDCG@20 comparison across different retrievers and datasets. With `Route Oracle` as an analytical way for mixture, we can get better performance than large language model-based retrievers.

performing retriever, assuming that the mixture weights just rule out useless retrievers. We denote these analytical results as `Route Oracle`. Comparing `Route Oracle` with state-of-the-art LLM retrievers with 7B parameters, as shown in Table 2, we see that the former achieves consistently better performance than GritLM (overall +13.5%), which suggests the potential gain from the mixture.

4 Mixture of Retrievers.

Section 3.2 suggests that mixing small retrievers has the potential to outperform large ones. However, `Route Oracle` operates under a simplified setting. In practice, identifying the ideal retriever to route to is nontrivial, and naive routing overlooks the potential gains from leveraging signals across multiple retrievers. To address this, we propose a more practical framework: `MoR` (Mixture of Retrievers), which incorporates a diverse set of signals capturing interactions between queries, retrievers, and documents—without relying on groundtruths.

The desiderata. Recall that a retriever $\mathcal{R}_i \in L_{\mathcal{R}}$ assigns a relevance score $s_i(q, d_j)$[1] to each document d_j, given a query q and corpus $\mathcal{D} = \{d_j\}_{j=1}^{|\mathcal{D}|}$. This results in a full corpus-sized vector $\mathbf{s}_i \in \mathbb{R}^{|\mathcal{D}|}$. Each query is then sent to all retrievers, resulting in N relevance scores per document d_j, where $N = |L_{\mathcal{R}}|$ is the number of retrievers.

Concurrently, we compute a scalar weight for each retriever $\mathcal{R}_i$, using a weight allocation function $f(q, \mathcal{R}_i, \mathcal{D})$, which estimates the *effectiveness* of $\mathcal{R}_i$ for the given query. These weights are then used to compute an adjusted relevance score $\tilde{s}(q, d_j)$ for each document d_j via a weighted sum over all retrievers:

$$\tilde{s}(q, d_j) = \sum_{i=1}^{N} f(q, \mathcal{R}_i, \mathcal{D}) s_i(q, d_j),$$

followed by re-ranking of the documents $\{d_j\}$ based on the adjusted scores.

A key advantage of this approach is that, due to the zero-shot nature of our geometry-based scores in the embedding space, it naturally supports a *plug-and-play* integration of diverse retrievers, including human information sources that provide ranked documents, as long as appropriate sentence embedding models are available. We explore the collaborative setup further in Section 5.2.

4.1 Comprehensive retriever coverage

Inspired by Chen et al. (2023b) and Cai et al. (2024), retrieval indices with different granularities may provide different performance to retrievers, e.g., sub-question index may be paired better with keyword matching, while sentence-level index may work better with semantic embedding-based retrievers. Besides the list of retrievers we introduce in Section 1, for each retriever $\mathcal{R}_i \in L_{\mathcal{R}}$, we also consider its variants with various kinds of granularity

[1]The scores are normalized to [0,1] for each retriever.

matching. We denote this retriever expansion as a **deep fusion** of retrievers.

Specifically, we consider four variants of retrievers with different indices: $R^{\text{q-d}}$ (original questions and documents), $R^{\text{q-p}}$ (questions and propositions), $R^{\text{sq-p}}$ (sub-questions and passage), and $R^{\text{sq-p}}$ (sub-questions and propositions), following the original notations. To acquire aligned semantic units, we utilize *propositioner* released by Chen et al. (2023b) to break down both queries and documents into atomic units, i.e., sub-questions and propositions (atomic short sentences), respectively.[2] This means that we extend the number of retrievers by four times ($4N$), with no adaptation on the retrievers needed. We include details and examples of the proposition decomposition in Appendix A.2.

4.2 Weighing retrievers' effectiveness

We explore two complementary weight allocation methods to form $f(q, \mathcal{R}_i, \mathcal{D})$: (i) *Pre-retrieval*: we estimate which retrievers to focus more on by comparing the query embedding to the document embedding centroids generated by each retriever; and (ii) *Post-retrieval*: we estimate the effectiveness of each retriever by comparing its ranked retrieved documents to the overall corpus distribution.

Pre-retrieval signals. We define pre-retrieval signals as measures of how likely the retriever can perform well on a query before seeking the top relevant documents from the whole corpus. This is closely related to the *when-to-retrieve* problem, where previous work studies from the angles of query token occurrences (Mallen et al., 2023) and model familiarity (Thrust, Zhao et al., 2023). We extend previous cluster-based familiarity analysis among queries with LLM generators (i.e., readers) to the query-document relations with retrievers. Specifically, for the embedding space of each retriever $\mathcal{R}_i$, we use KMeans Clustering to build K clusters $\{C_1, ..., C_k\}$ over the corpus $\mathcal{D}$.[3]

We design the new pre-retrieval signal as the sum of the vectorized distance from the query vector ($\vec{q}$) to the cluster centroid vectors ($\{\vec{m}_1, ..., \vec{m}_k\}$) (weighed by their sizes). Then, when the vectorized distance is large, it means that the query vector is distant from all clusters or with similar distance to distant clusters. In the former case, the query is likely an outlier for $\mathcal{R}_i$, whereas the latter suggests that the retriever can not categorize the query into any type of documents. As a result, we down-weight $\mathcal{R}_i$. On the other hand, a small distance suggests that the query clearly belongs to one type of document, where we up-weight $\mathcal{R}_i$ with the reciprocal of the distance. If we denote the retriever encoded query as $\vec{q}$, the pre-retrieval signal is defined as follows:

$$V_{\text{pre}}(q, \mathcal{R}_i, \mathcal{D}) \triangleq \left\| \sum_{k=1}^{K} \frac{|\mathcal{C}_k|}{K} \cdot \hat{v}_k \cdot \frac{1}{\|\vec{v}_k\|_2^2} \right\|,$$

where $\vec{v}_k = \vec{m}_k - \vec{q}$, the vectorized distance pointing from the query vector to the centroid embedding m_k of the k'th cluster (with a corresponding unit vector $\hat{v}_k$ indicating the direction), and $|\mathcal{C}_k|$ being the size of the k'th cluster. This effectively captures the semantic proximity between the query and the corpus accessible to the retriever and signals how closely the query embedding aligns with the dominant regions of the corpus under $\mathcal{R}_i$, with higher values suggesting greater familiarity.

Post-retrieval signals. We define post-retrieval signals as measures of how likely the retrieved results are correct, building on a rich line of work in the query performance prediction literature (Diaz, 2007; Long and Chang, 2014; Roitman, 2017; Singh et al., 2023; Arabzadeh et al., 2024). To preserve the zero-shot nature of our approach, we use the Moran coefficient (Diaz, 2007), denoted as $I_{\text{Moran}}(q, \mathcal{R}_i, \mathcal{D})$. This coefficient produces a scalar value that quantifies the correlation among the retrieved documents and has been shown to correlate with retrieval performance. It builds on the cluster-hypothesis (Jardine and van Rijsbergen, 1971), which posits that closely related documents tend to be relevant to the same query. Therefore, a higher coefficient indicates a greater likelihood that the retrieved documents are relevant to the query.

In addition to measuring relevance among retrieved documents using the Moran coefficient, we also assess their relevance with respect to the entire corpus. Similar to our pre-retrieval signals, we further extend the sum of the vectorized distance from query-document comparison to direct document-document comparison. Suppose that for a query q, the top-ranked retrieved documents are $\mathcal{D}_q$,

$$V_{\text{post}}(q, \mathcal{R}_i, \mathcal{D}) \triangleq \left\| \frac{1}{|\mathcal{D}_q|} \sum_{n=1}^{|\mathcal{D}_q|} V_{\text{pre}}(d_n, \mathcal{R}_i, \mathcal{D}) \right\|,$$

where empirically, we set $|\mathcal{D}_q| = 20$.

[2] For a sentence, *Alice and Bob had coffee together*, the propositions can be *Alice had coffee* and *Bob had coffee*. More details and efficient proposition representation extraction methods can be found in (Chen et al., 2023a).

[3] We choose K to be $\max(\text{ceil}(\sqrt[4]{|\mathcal{D}|}), 3)$.

	NFCorpus		SciDocs		SciFact		SciQ		**Average across datasets**	
	NDCG@5	NDCG@20	NDCG@5	NDCG@20	NDCG@5	NDCG@20	NDCG@5	NDCG@20	NDCG@5	NDCG@20
Unsupervised Retrievers										
BM25	37.8	30.7	14.8	19.9	64.7	69.2	91.9	92.2	52.3	53.0
SimCSE	16.2	13.3	7.6	9.7	27.1	31.2	62.3	67.3	28.3	30.4
Contriever	42.2	34.9	13.5	18.5	64.5	68.5	67.2	70.0	46.9	48.0
Supervised Dense Retrievers										
DPR	25.1	20.7	7.3	10.4	31.8	37.7	60.6	64.1	31.2	33.2
ANCE	19.9	24.4	9.3	13.1	41.5	45.3	66.4	69.1	34.3	38.0
TAS-B	42.3	34.1	13.8	19.3	60.1	65.6	84.8	86.3	50.3	51.3
MPNet	45.6	38.7	21.3	30.3	64.9	69.4	68.0	71.9	50.0	52.3
GTR	42.1	34.1	13.6	18.9	58.3	62.2	83.3	84.4	49.3	49.9
Mixture of Retrievers										
MoR-pre	47.7	40.4	20.9	27.5	68.7	72.8	91.4	91.6	57.2	58.1
MoR-post	**48.0**	**40.5**	**21.5**	**28.1**	68.9	73.2	**92.7**	**92.8**	**57.8**	**58.7**
LLM-based Retrievers (7b parameters)										
RepLLaMA	39.8	36.4	11.9	18.2	72.5	74.1	63.3	65.9	46.9	48.7
GritLM	47.7	38.8	20.3	27.7	**76.9**	**79.8**	78.4	79.7	55.3	56.5

Table 3: Performance comparison of different retrievers across datasets using NDCG@5 and NDCG@20 metrics. *Avg.* denotes the macro-average across the tasks. **Bold** indicates the best performing rows. MoR variants achieve better overall performance than their component retrievers, as well as non-component LLM-based retrievers.

4.3 Parametric Combination.

Upon acquiring the individual signals, we design two types of per-query per-retriever weight allocation methods. For pre-retrieval MoR-pre, $f_{\text{pre}}(q, \mathcal{R}_i, \mathcal{D}) = V_{\text{pre}}(q, \mathcal{R}_i, \mathcal{D})$. For post-retrieval MoR-post, we consider the parametric combinations of the signals, i.e., $f_{\text{post}}(q, \mathcal{R}_i, \mathcal{D}) = a \cdot V_{\text{pre}} + b \cdot I_{\text{Moran}} + c \cdot V_{\text{post}}$, where coefficients (a, b, c) are hyperparameters. Empirically, we select the final set of (a, b, c) for MoR-post as $(0.1, 0.3, 0.6)$. We use the same set of a, b, c for all queries. A query-specific weighting variant is in Appendix A.8. We also note that the optimal sets can vary across queries, where query-specific coefficients can constitute an interesting future investigation. More details are in our Limitations (Section 6).

5 Experiments and Analyses

To assess the MoR usefulness, we first compare the retrieval and RAG performance of MoR with various supervised and unsupervised retrievers in Section 5.1, using retrievers in Section 3.1.[4] Then, we experiment on how the mixture framework can help incorporate specialized non-oracle human information sources (Section 5.2). We further study MoR efficiency in Section 5.3. Details of the list retrievers included in MoR are in Section 3.1.

[4]RepLLaMA and GritLM are not considered as components of $L_{\mathcal{R}}$ for MoR due to efficiency reasons.

5.1 Main results: MoR is Effective

MoR improves retrieval performance. From Table 3, comparing MoR with their component retrievers, we can observe that our zero-shot signals achieve good performance on combining the query-document scores from different retrievers. Across different tasks, MoR-pre achieves improved performance over various unsupervised and supervised retrievers, which demonstrates the significance of our proposed V_{pre} without searching the most similar documents from the corpus. Through considering I_{Moran} and V_{post}, MoR-post achieves even better performance than the pre-retrieval variant, with a relative 10.8% and 12.2% performance improvement on NDCG@20 over the best unsupervised and supervised components, respectively.

With further comparison to LLM-based retrievers, we can observe that a mixture of smaller retrievers can surpass the performance of large ones, with an overall +3.9% better relative NDCG@20 improvement over GritLM. Across different tasks, MoR-post achieves better performance than GritLM on NFCorpus, SciFact, and SciQ with 5 or 20 chunks considered. For SciFact, MoR-post achieves better performance than its components and comparable performance to GritLM.

We show detailed qualitative examples in Appendix A.5 in the appendix to reveal further details on how MoR works, e.g., selecting the correct one when most retrievers fail or ignoring the wrong output from the overall best-performing retriever.

	SciFact		SciQ	
	EM@1	EM@3	EM@1	EM@3
BM25	67.4	71.5	64.1	68.8
MPNet	66.9	75.6	57.0	61.3
RepLLaMA	45.9	65.7	56.2	64.1
GritLM	66.9	77.3	61.7	66.8
MoR-pre	68.0	74.4	62.9	67.2
MoR-post	**72.9**	**77.9**	**66.8**	**68.8**

Table 4: Retrieval augmented generation performance on SciFact and SciQ with top-1 (EM@1) or 3 (EM@3) chunks fed into the reader model. EM denotes using Exact Match as the metric. **Bold** indicates the best performing row.

Domain Weights	Medicine	Psychology	CS	Eng.
Medicine Expert	**0.6**	0.0	0.0	0.0
Psychology Expert	0.0	**0.8**	0.0	0.0
CS Expert	0.0	0.1	**0.8**	0.1
Engineering Expert	0.2	0.0	0.1	**0.7**

Table 5: Averaged weights of V_{post} assigned to each simulated human expert on queries from different domains. Simulated experts can conduct Oracle retrieval on their corresponding domains. **Bold** denotes the highest weights for each query domain (column).

MoR improves RAG performance. Besides the gain on retrieval performance. We further validate the impact of MoR on RAG with SciFact and SciQ that have downstream generation tasks, i.e., fact-checking and question answering, respectively. To do so, we use the standard RAG pipeline to feed the doc chunks retrieved into a reader model Llama-3-8B-Instruct (Touvron et al., 2023b). We use Exact Match (EM@K) to measure the retrieval-augmented generation performance, where K denotes the number of retrieved chunks considered.

From Table 4, we can observe that, similar to the retrieval performance, MoR shows consistent performance improvement over baselines, which demonstrates the effectiveness of MoR on RAG even without utilizing any signals from the final generation beforehand to assign weights. Yet, post-presentation signals from downstream generation, e.g., accuracy, can still be important future work beyond our current scope. Additionally, results for Open-domain Natural-Questions are in Appendix A.9.

5.2 Human as a Retriever

In our main experiments, we mainly consider the mixture of sparse and dense retrievers, such as BM25 and DPR. However, for complex scientific domain retrieval, human experts shall also be considered as an important source of information. In this section, we conduct a stress test of MoR to explore the potential to generalize to all kinds of information sources, including using humans as retrievers. Specifically, we simulate human experts using four domains from the original SciDocs splits[5], including Medicine, Psychology, Computer Science (CS), and general Engineering (Eng.). We construct 4 corresponding human experts, where each expert can conduct Oracle retrieval on its own domain (gold documents are ranked top, and the rest and ranked with their relevance to the gold documents). For queries from other domains, the simulated experts will output random ranks. We use MPNet to encode the ranked documents into their semantic representation.

NDCG@20	Medicine	Psychology	CS	Eng.
GritLM	42.3	48.1	20.5	42.2
MoR-post	44.3	44.4	20.7	43.2
Human Experts	71.2	53.5	40.0	66.2
MoR+Humans	87.2	91.5	94.1	94.3

Table 6: Performance of each domain with GritLM, MoR, and MoR+Humans. Each human expert can conduct Oracle retrieval on the corresponding domain, but random retrieval for others. *Humans Experts* denotes the retrieval performance when we assign equal weights to each one's ranks.

MoR assigns reasonable weights to simulated human retrievers. With the setting above, we study whether our V_{post} can delegate reasonable weights to the human experts. From Table 5, we can observe that experts are consistently weighted high in their corresponding domains (which means their ranks will be counted more in the final MoR ranks) and low in non-expert domains. The reasonable weights highlight the effectiveness of V_{post} in a controlled setting, and show that our method has the potential to help estimate human trustworthiness from a given corpus.

MoR achieves improved performance to simulated human retrievers. We see from Table 6 that without simulated human experts, MoR achieves on par performance with GritLM on SciDocs retrieval task for these specific domains. However, through delegating reasonable weights to the human experts and include their ranks into consid-

[5] https://github.com/allenai/scidocs. General engineering denotes non-CS engineering topics.

Best of	Retrievers	NDCG@20
2	Contriever, GTR	92.6
3	Contriever, GTR. MPNet	92.8
4	SimCSE, DPR, Contriever, GTR	92.9

Table 7: Results mixing a subset of X retrievers for SciQ, with `MoR-post`. Mixing the best 2 achieves comparable results to our original mixture of 8 in Table 3.

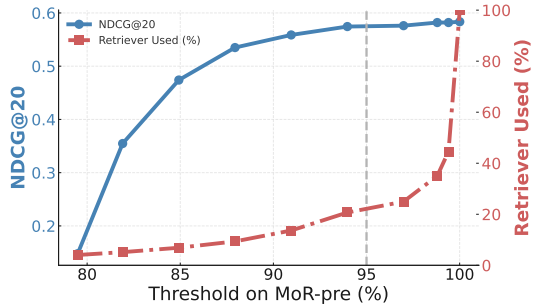

Figure 3: Performance and equivalent retriever used (%) at different thresholds on `MoR-pre` weights. Threshold (%) at 95 percentile between min and max weights allows `MoR` to maintain most performance with only around 20% equivalent retriever use.

eration, `MoR` achieves consistent best performance across all domains, outperforms aggregating the human experts (+58.9% relatively), which shows its potential of including non-oracle humans as information sources in collaborative tasks.

5.3 Efficiency

In addition to evaluating performance, we also consider efficiency as a key metric for assessing `MoR`. Following standard RAG conventions, corpus embeddings are pre-computed during offline preparation. As a result, the primary computational cost at test time arises from encoding the input queries and searching for the most relevant corpus entries — Both increases linearly with the number of dense retrievers we include. To improve the `MoR`'s efficiency, we pose the question: *Which retrievers should we include?*

As a stress test, we exhaustively enumerated all possible subsets of X retrievers drawn from $L_{\mathcal{R}}$, and examine the performance of the best combinations. As shown in Table 7, mixing just two retrievers can already yield performance comparable to mixing all eight. Notably, the best-performing pair is *not* composed of the individually top-performing retrievers (MPNet and BM25, per Table 3). This suggests that subset selection can significantly improve `MoR` efficiency without sacrificing much performance; However, effective selection should prioritize complementarity over absolute performance.

	Granularity Merge	Retriever Merge	Avg. ND@20
`MoR`	max	mean	46.3
`MoR`	mean	mean	35.2
`MoR`	mean	V_{pre}	50.7
`MoR`	none	$V_{\text{pre}}, I_{\text{Moran}}, V_{\text{post}}$	56.0
`MoR-pre`	V_{pre}		58.1
`MoR-post`	$V_{\text{pre}}, I_{\text{Moran}}, V_{\text{post}}$		58.7

Table 8: Ablation studies with retrieval performance of Avg. NDCG@20 metrics. Granularity Merge denotes the way to merge the scores of the same retriever operated in different granularities. Retriever Merge denotes the way to merge scores from different retrievers.

Building on this, we explore a natural extension to `MoR`: pre-rejecting certain retrievers using `MoR-pre` weights, calculated *before* observing actual query-document interactions. Concretely, we normalize `MoR-pre` scores into percentiles and apply a rejection threshold, discarding low-ranking retrievers accordingly. Figure 3 shows this early rejection strategy is effective: at a 95th percentile threshold, `MoR` maintains strong performance while using, on average, around only 20% of retrievers per query. Full latency numbers are in Appendix A.7.

5.4 Ablation Study

In Table 8, we further validates the design choices of `MoR` with ablations. Specifically, we consider variants of `MoR` with different treatments on the weight delegation on various retrievers (Retriever Merge), as well as their variants on different granularities (Granularity Merge). From the comparison of the average performance across our task suite, Table 8 presents that the signals we fetched show significantly better performance than adhoc mean or max over the retriever scores for each query. Besides, the no granularity merge variant shows the effectiveness of including the deep fusion. Further comparison of individual components of our method as well as other alternatives, e.g., Reciprocal Rank Fusion (RRF), can be found in Appendix A.4.

6 Conclusion

In this paper, we propose to construct a mixture of retrievers (`MoR`) to improve the retrieval generalizability on diverse and complex retrieval tasks, leveraging the comparative advantages among small-

scaled retrievers. To do so, we propose to use deep fusion considering multi-granularities, as well as design various pre-/post-retrieval signals to weight the outputs of each retriever for each query. Experiments on various tasks and settings validate the consistent and robust performance improvement of MoR compared to its component retrievers and SOTA LLM-based retrievers. Extensive analysis further sheds light on how MoR can potentially incorporate human information sources and be implemented with improved efficiency. We open-source our code at https://github.com/Josh1108/MixtureRetrievers.

Limitations

Post-presentation Signals. In our main experiments, we design various effective signals before and after conducting retrieval. However, there is one source of signals that can be an important future direction - the end-to-end performance after utilizing the retrieved documents, e.g., the exact match performance on question answering or the user satisfaction. We denote such signals as post-presentation signals, which can potentially extend the design from the retrieval perspective to the RAG (Jiang et al., 2023) or agentic retrieval (Li et al., 2025) perspectives.

Supervised Methods. In this paper, we focus on proposing the MoR architecture and design of unsupervised signal sources to allow potential extensiveness to future retrievers. However, there is a zoom for potential supervised methods at different stages of MoR. First, intuitively, all the per query per retriever weight can be learned through a neural network, considering q, R, D', D as the inputs given a small set of training data, with the pre-computed optimal weights. Besides, our current parametric combination uses one universal set of coefficients for different signals. Another neural network can be used to compute the coefficients per retriever or per query with the query embeddings as the inputs, which can potentially help model the different kinds of complexity among queries, i.e., the complexity can come from comprehending the query itself (captured by layer-variance) or the lack of good entries (captured by post-retrieval signals).

Acknowledgments

The authors thank Tong Chen, Yuhao Zhang, Haoyang Wen, Hongming Zhang, Sihao Chen, Ben Zhou, and Lexin Zhou for their insights into design and evaluation choices. The authors also thank the constructive discussions with colleagues from CMU WInE Lab. Xinran Zhao is supported by the ONR Award N000142312840. To Eun Kim is supported by NSF grant 2402874. This work is supported by the OpenAI Research Credit program, the Amazon AI Research Gift Fund, and the Gemma Academic Program GCP Credit Award. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsors.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Negar Arabzadeh, Chuan Meng, Mohammad Aliannejadi, and Ebrahim Bagheri. 2024. Query performance prediction: Techniques and applications in modern information retrieval. In *Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, SIGIR-AP 2024, page 291–294, New York, NY, USA. Association for Computing Machinery.

Jaime Arguello and Fernando Diaz. 2013. *Relevance Ranking of Vertical Search Engines*, chapter Vertical Selection and Aggregation. Elsevier.

Jaime Arguello and 1 others. 2017. Aggregated search. *Foundations and Trends® in Information Retrieval*, 10(5):365–502.

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*.

Vera Boteva, Demian Gholipour, Artem Sokolov, and Stefan Riezler. 2016. A full-text learning to rank dataset for medical information retrieval. In *Advances in Information Retrieval: 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20–23, 2016. Proceedings 38*, pages 716–722. Springer.

Fengyu Cai, Xinran Zhao, Tong Chen, Sihao Chen, Hongming Zhang, Iryna Gurevych, and Heinz Koeppl. 2024. MixGR: Enhancing retriever generalization for scientific domain through complementary granularity. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10369–10391, Miami, Florida, USA. Association for Computational Linguistics.

James P Callan, Zhihong Lu, and W Bruce Croft. 1995. Searching distributed collections with inference networks. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 21–28.

Jamie Callan. 2002. Distributed information retrieval. In *Advances in information retrieval: recent research from the center for intelligent information retrieval*, pages 127–150. Springer.

Hsinchun Chen, Haiyan Fan, Michael Chau, and Daniel Zeng. 2001. Metaspider: Meta-searching and categorization on the web. *Journal of the American Society for Information Science and Technology*, 52(13):1134–1147.

Sihao Chen, Hongming Zhang, Tong Chen, Ben Zhou, Wenhao Yu, Dian Yu, Baolin Peng, Hongwei Wang, Dan Roth, and Dong Yu. 2023a. Sub-sentence encoder: Contrastive learning of propositional semantic representations. *arXiv preprint arXiv:2311.04335*.

Tong Chen, Hongwei Wang, Sihao Chen, Wenhao Yu, Kaixin Ma, Xinran Zhao, Dong Yu, and Hongming Zhang. 2023b. Dense x retrieval: What retrieval granularity should we use? *arXiv preprint arXiv:2312.06648*.

Eunsol Choi, Jennimaria Palomaki, Matthew Lamm, Tom Kwiatkowski, Dipanjan Das, and Michael Collins. 2021. Decontextualization: Making sentences stand-alone. *Transactions of the Association for Computational Linguistics*, 9:447–461.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, and 1 others. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel Weld. 2020. SPECTER: Document-level representation learning using citation-informed transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2270–2282, Online. Association for Computational Linguistics.

Gordon V Cormack, Charles LA Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 758–759.

Zhuyun Dai, Yubin Kim, and Jamie Callan. 2017. Learning to rank resources. In *Proceedings of the 40th International ACM SIGIR conference on research and development in information retrieval*, pages 837–840.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Fernando Diaz. 2007. Performance prediction using spatial autocorrelation. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, page 583–590, New York, NY, USA. Association for Computing Machinery.

Fernando Diaz, Mounia Lalmas, and Milad Shokouhi. 2010. From federated to aggregated search. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 910–910.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Eric J Glover, Steve Lawrence, William P Birmingham, and C Lee Giles. 1999. Architecture of a metasearch engine that supports user information needs. In *Proceedings of the eighth international conference on Information and knowledge management*, pages 210–216.

Rachid Guerraoui, Anne-Marie Kermarrec, Diana Petrescu, Rafael Pires, Mathis Randl, and Martijn de Vos. 2025. Efficient federated search for retrieval-augmented generation. In *Proceedings of the 5th Workshop on Machine Learning and Systems*, EuroMLSys '25, page 74–81, New York, NY, USA. Association for Computing Machinery.

Charles R. Harris, K. Jarrod Millman, Stéfan van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, and 7 others. 2020. Array programming with numpy. *Nature*, 585:357–362.

Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently teaching an effective dense retriever with balanced topic aware sampling. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 113–122.

John D Hunter. 2007. Matplotlib: A 2d graphics environment. *Computing in science & engineering*, 9(03):90–95.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. 1991. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87.

N. Jardine and C.J. van Rijsbergen. 1971. The use of hierarchic clustering in information retrieval. *Information Storage and Retrieval*, 7(5):217–240.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong Park. 2024. Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity. 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 7036–7050, Mexico City, Mexico. Association for Computational Linguistics.

Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992, Singapore. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020a. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020b. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6769–6781. Association for Computational Linguistics.

Ekaterina Khramtsova, Shengyao Zhuang, Mahsa Baktashmotlagh, Xi Wang, and Guido Zuccon. 2023. Selecting which dense retriever to use for zero-shot search. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, SIGIR-AP '23, page 223–233, New York, NY, USA. Association for Computing Machinery.

To Eun Kim and Fernando Diaz. 2025. Ltrr: Learning to rank retrievers for llms. *Preprint*, arXiv:2506.13743.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Langchain. 2025. Langchain ensembleretriever. Accessed: 04-May-2025.

Hyunji Lee, Luca Soldaini, Arman Cohan, Minjoon Seo, and Kyle Lo. 2024. Routerretriever: Exploring the benefits of routing over multiple expert embedding models. *arXiv preprint arXiv:2409.02685*.

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.

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. 2025. Search-o1: Agentic search-enhanced large reasoning models. *CoRR*, abs/2501.05366.

Bo Long and Yi Chang. 2014. *Relevance Ranking for Vertical Search Engines*, 1st edition. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. 2023. Fine-tuning llama for multi-stage text retrieval.

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, Toronto, Canada. Association for Computational Linguistics.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.

Feiteng Mu, Yong Jiang, Liwen Zhang, Liuchu Liuchu, Wenjie Li, Pengjun Xie, and Fei Huang. 2024. Query routing for homogeneous tools: An instantiation in the RAG scenario. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages

10225–10230, Miami, Florida, USA. Association for Computational Linguistics.

Niklas Muennighoff, Hongjin Su, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. 2024. Generative representational instruction tuning.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*.

Giang Ngo, Rodney Beard, and Rohitash Chandra. 2022. Evolutionary bagging for ensemble learning. *Neurocomputing*, 510:1–14.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, and Yinfei Yang. 2022. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9844–9855, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, and 2 others. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

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

Haggai Roitman. 2017. An enhanced approach to query performance prediction using reference lists. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, page 869–872, New York, NY, USA. Association for Computing Machinery.

Kunal Sawarkar, Abhilasha Mangal, and Shivam Raj Solanki. 2024. Blended rag: Improving rag (retriever-augmented generation) accuracy with semantic search and hybrid query-based retrievers. In *2024 IEEE 7th International Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 155–161.

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Q. Tran, Yi Tay, and Donald Metzler. 2022. Confident adaptive language modeling. In *Advances in Neural Information Processing Systems*.

Noam Shazeer, *Azalia Mirhoseini, *Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*.

Ashutosh Singh, Debasis Ganguly, Suchana Datta, and Craig McDonald. 2023. Unsupervised query performance prediction for neural models with pairwise rank preferences. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, page 2486–2490, New York, NY, USA. Association for Computing Machinery.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *arXiv preprint arXiv:2004.09297*.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023b. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Andrew Trotman, Antti Puurula, and Blake Burgess. 2014. Improvements to bm25 and language models examined. In *Proceedings of the 2014 Australasian Document Computing Symposium*, ADCS '14, page 58–65, New York, NY, USA. Association for Computing Machinery.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, Online. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. 2017. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 94–106, Copenhagen, Denmark. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and 1 others. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv preprint*, abs/1910.03771.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*.

Woongyeong Yeo, Kangsan Kim, Soyeong Jeong, Jinheon Baek, and Sung Ju Hwang. 2025. Universalrag: Retrieval-augmented generation over multiple corpora with diverse modalities and granularities. *arXiv preprint arXiv:2504.20734*.

Tony Z. Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models.

Xinran Zhao, Hongming Zhang, Xiaoman Pan, Wenlin Yao, Dong Yu, and Jianshu Chen. 2023. Thrust: Adaptively propels large language models with external knowledge. *arXiv preprint arXiv:2307.10442*.

A Appendix

A.1 Experiment Details

Statistics of the data. Table 9 presents the statistics of four datasets used in our experiments, together with the statistics of the decomposed queries and documents, i.e., sub-queries and propositions. Specifically, due to the max-length requirement for some dense retrievers such as DPR (Karpukhin et al., 2020a), we split one document in the original dataset into several chunks containing a maximum of 128 words. In this way, we can avoid the loss of information caused by context overflow. The retrieved chunk serves as a reference to locate the corresponding document in the original dataset for evaluation.

Infrastructure. We conduct experiments on a Google Cloud Platform instance equipped with 8×NVIDIA L4 GPUs, each with 24 GB of memory.

Setup and Hyperparameters. For retrieval augmented generation, we use a temperature of 0.1 and top_p value of 0.7 across all tasks.

A.2 Propositionizer details and decompositions

Propositioner (Chen et al., 2023b)[6] is an off-the-shelf model for query and document decomposition. Using Wikipedia as the dataset, this model distills GPT-4's (Achiam et al., 2023) decomposition capacity into Flan-T5-Large (Chung et al., 2024).

Propositioner breaks down queries and documents into fundamental components—subqueries and propositions, respectively. Each proposition (or subquery) is expected to satisfy the following three key criteria (Min et al., 2023):

- It should express a distinct semantic unit, contributing to the overall meaning when considered with others.
- It must be atomic and indivisible.
- Following Choi et al. (2021), each proposition should be self-contained and contextually complete, incorporating all necessary information, such as resolved coreferences, for unambiguous interpretation.

The example of subqueries and propositions is listed in Figure 4.

[6] https://huggingface.co/chentong00/propositionizer-wiki-flan-t5-large

Query: Citrullinated proteins externalized in neutrophil extracellular traps act indirectly to perpetuate the inflammatory cycle via induction of autoantibodies.

- **Subquery-0**: Citrullinated proteins are externalized in neutrophil extracellular traps.
- **Subquery-1**: Citrullinated proteins act indirectly to perpetuate the inflammatory cycle.
- **Subquery-2**: The inflammatory cycle is perpetuated via induction of autoantibodies.

Document: In humans, RNA blot analysis revealed that Golli-MBP transcripts were expressed in fetal thymus, spleen, and human B-cell and macrophage cell lines, as well as in fetal spinal cord. These findings clearly link the expression of exons encoding the autoimmunogen/encephalitogen MBP in the central nervous system to cells and tissues of the immune system through normal expression of the Golli-MBP gene. They also establish that this genetic locus, which includes the MBP gene, is conserved among species, providing further evidence that the MBP transcription unit is an integral part of the Golli transcription unit and suggest that this structural arrangement is important for the genetic function and/or regulation of these genes.

- **Proposition-0:** The human myelin basic protein (MBP) gene is contained within a 179-kilobase transcription unit.
- **Proposition-1:** Golli-MBP transcripts are expressed in fetal thymus, spleen, human B-cell lines, macrophage cell lines, and fetal spinal cord.
- **Proposition-2:** Expression of MBP-encoding exons in the central nervous system is linked to immune-system cells and tissues through normal Golli-MBP expression.
- **Proposition-3:** The genetic locus that includes the MBP gene is conserved across species.
- **Proposition-4:** The MBP transcription unit is an integral part of the larger Golli transcription unit.
- **Proposition-5:** The structural arrangement of the MBP and Golli transcription units is important for the genetic function and/or regulation of these genes.

Figure 4: Example of query and document decomposition with *Propositioner*.

Statistic (#)	NFCorpus (Boteva et al., 2016)	SciDocs (Cohan et al., 2020)	SciFact (Wadden et al., 2020)	SciQ (Welbl et al., 2017)
Query	1 016	1 000	1 109	884
Multi-subquery queries	641	205	283	252
Subqueries	3 337	522	614	874
Documents	3 633	25 657	5 183	12 241
Propositions	67 110	351 802	87 190	91 635

Table 9: Statistics for the NFCorpus, SciDocs, SciFact, and SciQ datasets. We note that these statistics have been adjusted to prevent proposition/sub-question decomposition errors.

Model	HuggingFace Checkpoint	Params
SimCSE (Gao et al., 2021)	`princeton-nlp/unsup-simcse-bert-base-uncased`	110M
Contriever (Izacard et al., 2022)	`facebook/contriever`	110M
DPR (Karpukhin et al., 2020a)	`facebook/dpr-ctx_encoder-multiset-base` `facebook/dpr-question_encoder-multiset-base`	110M
ANCE (Xiong et al., 2021)	`castorini/ance-dpr-context-multi` `castorini/ance-dpr-question-multi`	110M
TAS-B (Hofstätter et al., 2021)	`sentence-transformers/msmarco-distilbert-base-tas-b`	66M
GTR (Ni et al., 2022)	`sentence-transformers/gtr-t5-base`	220M
MPNet (Song et al., 2020)	`sentence-transformers/all-mpnet-base-v2`	110M
RepLLaMA (Ma et al., 2023)	`castorini/repllama-v1-7b-lora-passage`	7B
GritLM (Muennighoff et al., 2024)	`GritLM/GritLM-7B`	7B

Table 10: Model checkpoints released on HuggingFace and model parameters. For DPR and ANCE, the parameter count is shared across the dual encoders.

RAG experiment details. The RAG templates used for SciFact and SciQ are listed below. For SciQ, we convert the multiple-choice questions into open-ended questions.

> Given the knowledge source: *context* \\n Question: *query* \\n Reply with one phrase. \\n Answer:

Since SciFact is a fact-checking task, we evaluate whether LLMs can accurately predict the relationship between a claim and a given context. The template used for SciFact is as follows:

> Context: {*context*} \\n Claim: {*query*} \\n For the claim, the context is supportive, contradictory, or not related? \\n Options: (A) Supportive (B) Contradictory (C) Not related \\n Answer:")

A.3 Extended Retriever Descriptions

We consider both sparse and various BERT-sized (Devlin et al., 2019) dense retrievers as candidates $L_{\mathcal{R}}$ to create our mixture of retrievers:

- BM25 (Trotman et al., 2014) is a traditional keyword matching based sparse retriever that shows good zero-shot performance (Thakur et al., 2021). We use TF-IDF vectors to conduct the vector space operations for BM25.
- SimCSE (Gao et al., 2021) employs a BERT-base (Devlin et al., 2019) encoder trained on self-supervised contrastive signals on Wikipedia sentences.
- Contriever (Izacard et al., 2022) is an unsupervised retriever evolved from a BERT-base encoder, contrastively trained on segments from unlabelled web and Wikipedia documents.
- DPR (Karpukhin et al., 2020a) is built with a dual-encoder BERT-base architecture, finetuned on a suite of open-domain datasets with labels, such as SQuAD (Rajpurkar et al., 2016).
- ANCE (Xiong et al., 2021) extend DPR with a training scheme of Approximate Nearest Neighbor Negative Contrastive Estimation (ANCE).
- TAS-B (Hofstätter et al., 2021) is a dual-encoder BERT-base model distilled from ColBERT on MS MARCO (Bajaj et al., 2016).
- GTR (Ni et al., 2022) is a T5-base encoder, focusing on generalization, pre-trained on unlabeled QA pairs, and fine-tuned on labeled data including MS MARCO.
- MPNet (Song et al., 2020) is a BERT-alike model with advanced pre-training strategy. We use it as a sentence transformer to serve as a retriever

following (Reimers and Gurevych, 2019).

As depicted in the descriptions, the above retrievers vary in parameter sizes, backbone architectures, and training signals. In addition, we also set a competitive performance reference with various large-language-model-based retrievers with 7 billion parameters.

- RepLLaMA (Ma et al., 2023) extends the dual-encoder retriever training pipeline of DPR to LLaMA (Touvron et al., 2023a), which demonstrates advanced performance and long-context generalizability.
- GritLM (Muennighoff et al., 2024) proposes to jointly train language models with generative and embedding tasks, which leads to great performance on MTEB (Muennighoff et al., 2022).

Model Huggingface checkpoints and size. Table 10 lists model checkpoints on Huggingface and the model sizes. Our experiments cover modern retrievers from different architectures (e.g., BERT (Gao et al., 2021) or LLaMA (Ma et al., 2023)) and a wide range of model size, from 66M to 7B.

A.4 Extended design choice ablation

In Section 5.1, we show the performance of MoR-pre and MoR-post, which are parametric combinations of various signals extracted from queries and documents. In this section, we take a closer look at the individual components used in MoR, i.e., V_{pre}, Moran, and V_{post}, as well as other baseline signals. For V_{post}, we also compare with the version without deep fusion of multiple granularities (denoted as no D.F.). Following our intuition and design in Section 4, for pre-retrieval signals, we consider:

- Performance Normalization (Perf. Norm.): We consider the simplest baseline on a development set that contains 100 queries to compute the retrieval performance of each retriever and weight each of them with the normalized scores (higher performance indicates higher weights).
- Layer-wise Variance (Layer Var.): motivated by (Schuster et al., 2022), we consider the layer-wise variance of the retriever on the first, middle, and last layers at each neuron to indicate how much computation is required for the retriever to process a specific query. The reciprocal of the variance is then considered as the weight of each query per retriever.
- Clustering: We consider the variance of the cluster centroids of the vectors of the corpora that are extracted from each retriever. The reciprocal of variance is then used as the weight.
- Thrust (Zhao et al., 2023): we utilize the original implementation of the target query and 300 sample queries to compute the Thrust score representing the retriever familiarity of the queries. The scores are then normalized to serve as the weights, where lower familiarity scores denote less weight.

For post-retrieval signals, we consider:

- Score and Representation Variance: following (Khramtsova et al., 2023), we utilize the relations among the top retrieved documents to conduct the performance prediction. For score variance (Score Var.), we consider the variance of the scores of top-x retrieved documents. Similarly, for representation variance (Rep. Var.), we consider the variance of their embeddings. The reciprocal of variance is then used as the weight, where lower variance denotes higher weights.
- Reciprocal Rank Fusion (RRF; Cormack et al. 2009): following (Langchain, 2025), we consider RRF as a direct baseline to combine the document ranks of the retrievers.

From Table 11, we can observe that the signals we designed based on vectorized distances (V_{pre} and V_{post}) are already strong without parametric combination. On the other hand, the classic Moran Index contributes to the parametric combination but is not individually effective in the MoR context. Deep Fusion, similar to our analysis with *Oracle-deep*, improves MoR-post by 3.6% relatively.

For other signals, intuitive variance-based methods, e.g., Rep. Var., does not show good performance on delegating weights to different retrievers. Further calibration (Zhao et al., 2021) can be a potential direction to improve this sort of method. However, Thrust and RRF also present to be good signals that can be extracted before and after conducting retrieval. For the concision or the proposed MoR, we did not consider these signals in the parametric combination, yet further performance improvement is anticipated if we do so.

A.5 Qualitative Analysis

We qualitatively examine the behavior of `MoR` in Figure 12. The top block illustrates a scenario

	NFCorpus		SciDocs		SciFact		SciQ		Avg.	
	ND@5	ND@20	ND@5	ND@20	ND@5	ND@20	ND@5	ND@20	ND@5	ND@20
pre-retrieval signals										
Perf. Nor.	40.1	34.2	13.9	20.2	45.3	52.4	21.0	32.8	30.1	34.9
Layer Var.	40.0	34.2	14.0	20.3	34.5	43.8	20.6	32.4	27.3	32.7
Clustering	42.1	35.2	15.5	21.4	51.0	57.3	38.7	48.6	36.8	40.6
Thrust	42.8	35.6	15.7	21.3	56.9	61.8	70.0	74.3	46.3	48.2
V_{pre}	47.7	40.4	20.9	27.5	68.7	72.8	91.4	91.6	57.2	58.1
post-retrieval signals										
Score Var.	40.0	34.0	13.7	19.8	41.1	49.0	20.7	30.9	28.9	33.4
Rep. Var.	39.7	34.3	14.5	20.7	42.0	49.8	21.3	32.2	29.4	34.2
RRF	44.6	37.7	17.0	24.1	64.2	69.2	84.2	85.6	52.5	54.2
Moran	42.6	34.8	15.0	20.5	42.9	50.7	26.4	35.3	31.7	35.3
V_{post}	47.7	40.4	20.8	27.3	68.8	72.9	91.5	91.9	57.2	58.1
V_{post} no D.F.	47.3	39.7	20.3	27.1	69.3	73.2	82.2	83.9	54.8	56.0

Table 11: Performance comparison of different signals across datasets using NDCG@5 and NDCG@20 metrics. V_{post} no D.F. denotes our method without the deep fusion component.

where most retrievers fail to retrieve the correct passage, yet MoR successfully identifies the relevant one. The middle block highlights cases where MoR improves retrieval by effectively integrating signals from the most accurate retrievers. Finally, the bottom block demonstrates that MoR, while generally effective, cannot succeed when all base retrievers fail to retrieve the correct passage.

A.6 Extended Best Retriever Suite Analysis

In Section 5.3, we discussed the overall best retriever suite across tasks, given a different number of retrievers selected. We also notice that the best suite can be different for different tasks. In this section, we further show the best suite for each task. As shown in Table 7, we can observe that, similar to what we show in the main paper, the best suite of retrievers is not necessarily the best-performing retrievers. For example, on SciQ, the top-2 performing retrievers are MPNet and BM25 in terms of NDCG@20. However, the best suite is GTR (supervised) and Contriever (unsupervised). Such observations further validate our intuition in designing MoR - leveraging the comparative advantages among them. Similar findings generalize to other datasets as well.

On the other hand, comparing the retrieval performance of different numbers of *Best of X*, although the overall performance improve consistently with the a larger retriever list, the performance degradation is minor on some tasks. For example, on SciFact, the gap between *Best of 2* and *Best of 5* is 0.06 NDCG@5, which indicate that an efficient version of MoR to be deployed with a curated set of retrievers.

A.7 Runtime for MixGR

To see if MixGR could be deployed in a real world setting, we calculate the actual runtime of 100 queries for each retriever with an A6000 node. The total run time, as shown in Table 16 presents the advantage of MoR over GritLM (-18% actual runtime needed in seconds and better performance as shown in Table 3). We note that our further efficiency improvement methods discussed in Section 5.3 can further reduce 80% compute while maintaining 95% performance. On the other hand, if we host component retrievers of MoR in parallel in a single A6000 node, the run time for 100 queries can be reduced to 20 seconds.

A.8 Query Specific Combination of Signals

An important avenue for improvement lies in allowing the weighting coefficients (a, b, c) in our combination-of-retrievers setting to vary across queries. In the main system design, we adopt a fixed, uniform set of coefficients to ensure good off-the-shelf usability without requiring additional per-query tuning. While this choice emphasizes stability and general applicability, it raises the question of whether query-adaptive weights could further improve performance.

To explore this, we conducted an experiment comparing the current fixed coefficients with an oracle variant in which query-specific coefficients were selected via grid search. Results on NFCorpus are shown in Table 14.

The similar performance at NDCG@5 indicates

Query	Model	Recall@5	Top-5 Retrieved IDs
sciq-test_247	SimCSE	0	train_7285, test_58, train_1445, test_836, train_7766
	ANCE	0	test_58, train_2076, train_7766, train_8201, train_8099
	Contriever	0	train_7766, train_4806, train_8201, validation_576, train_7943
	TASB	0	test_34, test_983, train_8393, train_4806, train_3573
	MPNet	0	train_2076, train_7766, train_8201, test_836, test_58
	GTR	0	train_2076, test_836, train_3573, test_34, train_7766
	DPR	0	train_2076, train_8099, test_58, train_1944, validation_576
	BM25	1	test_247, test_836, train_6960, train_8201, train_10474
	MoR	1	test_836, test_247, train_4806, test_192, train_8201
sciq-test_0	SimCSE	1	test_0, train_3722, train_3328, train_6009, train_5098
	ANCE	1	test_0, train_6072, train_6454, train_4704, train_6736
	Contriever	1	test_0, train_2380, train_11507, train_465, train_7579
	TASB	0	train_443, train_7373, train_3464, train_9313, train_9608
	MPNet	1	train_4550, train_2886, test_0, train_6009, validation_355
	GTR	1	test_0, train_10570, train_4550, validation_355, train_3299
	DPR	1	test_0, train_2837, train_6072, validation_582, train_4485
	BM25	1	test_0, train_4704, train_10381, train_1544, train_443
	MoR	1	test_0, train_4550, train_11207, test_696, train_1997
sciq-test_143	SimCSE	0	train_677, train_11223, train_490, train_2716, train_5915
	ANCE	0	train_7701, train_5441, train_2716, validation_944, train_877
	Contriever	0	train_6186, train_6460, train_5114, train_10363, train_3753
	TASB	0	train_3753, train_2682, train_11467, train_9762, train_490
	MPNet	0	train_2340, test_669, train_2716, train_8417, train_8583
	GTR	0	train_11223, train_3673, train_3753, train_8417, train_2079
	DPR	0	train_7701, train_5441, train_2716, train_877, train_4268
	BM25	0	train_226, train_3753, train_11223, train_2716, train_3765
	MoR	0	train_7534, train_2716, train_490, train_998, train_3753

Table 12: Retrieval performance (Recall@5) and top-5 results per model for queries `sciq-test_247`, `sciq-test_0` and `sciq-test_143`, with correct hits highlighted in dark green.

that the current fixed coefficients already provide strong early precision. At the same time, the consistent gain in NDCG@20 under oracle tuning highlights the potential benefit of query-specific weighting. These findings suggest that while fixed coefficients offer a practical and robust default, developing tunable or adaptive strategies for (a, b, c) represents a promising direction for future work.

A.9 Application of MoR on Open Domain Datasets

We conducted a pilot study on Natural Questions (NQ) with 200 randomly sampled queries and 1000 associated chunks. Using identical settings and unchanged parameters, MoR consistently outperforms GritLM-7B, in line with our findings on scientific corpora, as shown in Table 17.

A.10 Ethical Statements

We foresee no ethical concerns or potential risks in our work. All of the retrieval models and datasets are open-sourced, as shown in Table 9 and Section A.3. The LLMs we applied in the experiments are also publicly available. Given our context, the outputs of LLMs are unlikely to contain harmful and dangerous information. The experiments in our paper are mainly on English.

A.11 Licences of Scientific Artifacts

We conclude the licenses of the scientific artifacts we used in Table 18. All artifacts are properly used following their original purposes.

A.12 Notation

We present a list of the notations we used in Table 19 for reference.

	Best of 2	Best of 3	Best of 4	Best of 5
Performance	NDCG@5/NDCG@20	NDCG@5/NDCG@20	NDCG@5/NDCG@20	NDCG@5/NDCG@20
Overall	56.9/57.9, MPNet, GTR	57.2/58.2 Contriever, GTR, MPNet	57.3/58.2 SimCSE, DPR, Contriever, GTR	57.2/58.1 SimCSE, MPNet, Contriever, GTR, DPR
SciQ	92.3/92.57, Contriever, GTR	92.64/92.77 Contriever, GTR, MPNet	92.77/92.90 SimCSE, DPR, Contriever, GTR	92.84/92.97 SimCSE, DPR, Contriever, GTR, MPNet
SciDocs	22.05/29.86 SimCSE, MPNet	22.09/29.60 SimCSE, DPR, MPNet	21.89/29.30 SimCSE, DPR, ANCE, MPNet	21.61/28.30 SimCSE, DPR, ANCE, Contriever, MPNet
NFCorpus	47.53/40.14 Contriever, MPNet	47.84/40.34 ANCE, Contriever, MPNet	47.90/40.31 ANCE, Contriever, MPNet, GTR	47.93/40.34 DPR, ANCE, Contriever, MPNet, GTR
SciFact	68.58/72.5 Contriever, MPNet	68.96/73.08 Contriever, MPNet, GTR	68.43/72.16 SimCSE, ANCE, Contriever, MPNet	68.64/72.87 DPR, SimCSE, ANCE, MPNet, Contriever

Table 13: Best suite of retrievers for `MoR-post` with different sizes of the retriever list.

Setting	NDCG@5	NDCG@20
Fixed coefficients (current)	48.0	40.5
Oracle query-specific coeff.	49.4	58.6

Table 14: Comparison of fixed vs. oracle query-specific coefficients on NFCorpus.

Method	MoR	GritLM
Time (seconds)	182.6	222.6

Table 15: Runtime for GritLM and MoR

Method	MoR	GritLM-7B
Time (seconds)	182.6	222.6

Table 16: End-to-end query latency including encoding and search.

Method	MoR	GritLM-7B
NDCG@20	86.9	82.7

Table 17: Pilot evaluation on Natural Questions (200 queries, 1000 chunks).

Artifacts/Packages	Citation	Link	License
SciFact	(Wadden et al., 2020)	https://huggingface.co/datasets/BeIR/scifact	cc-by-sa-4.0
SciDocs	(Cohan et al., 2020)	https://huggingface.co/datasets/BeIR/scidocs	cc-by-sa-4.0
SciQ	(Welbl et al., 2017)	https://huggingface.co/datasets/bigbio/sciq	cc-by-nc-3.9
NFCorpus	(Boteva et al., 2016)	https://huggingface.co/datasets/BeIR/nfcorpus	cc-by-sa-4.0
PyTorch	(Paszke et al., 2019)	https://pytorch.org/	BSD-3 License
transformers	(Wolf et al., 2019)	https://huggingface.co/transformers/v2.11.0/index.html	Apache License 2.0
numpy	(Harris et al., 2020)	https://numpy.org/	BSD License
matplotlib	(Hunter, 2007)	https://matplotlib.org/	BSD compatible License
vllm	(Kwon et al., 2023)	https://github.com/vllm-project/vllm	Apache License 2.0
LLaMA-3	(Touvron et al., 2023b)	https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct	LICENSE
SimCSE	(Gao et al., 2021)	https://huggingface.co/princeton-nlp/unsup-simcse-bert-base-uncased	MIT license
Contriever	(Izacard et al., 2022)	https://huggingface.co/facebook/contriever	License
DPR	(Karpukhin et al., 2020a)	https://huggingface.co/facebook/dpr-ctx_encoder-multiset-base	cc-by-nc-4.0
ANCE	(Xiong et al., 2021)	https://huggingface.co/castorini/ance-dpr-context-multi	MIT license
TAS-B	(Hofstätter et al., 2021)	https://huggingface.co/sentence-transformers/msmarco-distilbert-base-tas-b	Apache License 2.0
GTR	(Ni et al., 2022)	https://huggingface.co/sentence-transformers/gtr-t5-base	Apache License 2.0

Table 18: Details of datasets, major packages, and existing models we use. The curated datasets and our code/software are under the MIT License.

Notation	Description
q	user query
$\mathcal{D}$	corpus (stored retrievable items)
d_j	a document in a corpus
$\mathcal{R}_i(q, \mathcal{D})$	i'th retriever - simply $\mathcal{R}_i$
$L_{\mathcal{R}}$	set of retrievers
N	the number of retrievers ($\lvert L_{\mathcal{R}} \rvert$)
$s_i(q, d_j)$	query-document relevance score from the i'th retriever
$f(q, \mathcal{R}_i, \mathcal{D})$	MoR weight allocation function
$\tilde{s}(q, d_j)$	adjusted query-document relevance score after MoR weight aggregation

Table 19: Notation.