

000 001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 GRO-RAG: GRADIENT-AWARE RE-RANK OPTIMIZATION FOR MULTI-SOURCE RETRIEVAL-AUGMENTED GENERATION

Anonymous authors

Paper under double-blind review

ABSTRACT

Retrieval-Augmented Generation (RAG) systems often rely on information retrieved from heterogeneous sources to support generation tasks. However, existing approaches typically either aggregate all sources uniformly or statically select a single source, neglecting semantic complementarity. Moreover, they commonly employ re-ranking models to obtain Top-k documents, without accounting for actual contribution to generation objective. In this paper, we propose GRO-RAG, a training-free, gradient-aware re-ranking framework for multi-source RAG. Our method performs Top-k document selection by reading gradients from the language model, estimating each document’s contribution to the generation loss through a single backward pass. This enables re-ranking not by heuristic relevance, but by direct feedback from LLM’s generation objective. At the source level, we incorporate inter-source redundancy and query relevance to select source combination prior to re-ranking. Theoretically, we prove that this gradient-based Top-k selection approximates the optimal subset minimizing the generation loss, and aligns with minimizing the leave-one-out loss upper bound. Experiments across multi-source QA and open-domain generation tasks demonstrate consistent improvements in generation quality, highlighting the importance of generation-aware retrieval selection in multi-source RAG.

1 INTRODUCTION

Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) has emerged as a powerful paradigm for enhancing large language models (LLMs) by grounding their outputs in external knowledge. A typical RAG pipeline (Chen et al., 2017; Das et al., 2019) first retrieves a set of supporting documents from a corpus, and then conditions the generation process on both the query and the retrieved context. In practice, especially in open-domain and multi-hop settings, information is often distributed across multiple heterogeneous sources such as encyclopedias, web documents (Komeili et al., 2022; Dinan et al., 2019), or community forums. This gives rise to the challenge of multi-source retrieval, where the system must identify not only relevant documents, but also determine which sources to trust, combine, or ignore (Yan et al., 2024; Yao et al., 2023; Wang et al., 2024; Zhao et al., 2024a).

By integrating multiple retrieval sources, Multi-Source RAG mitigates the capability ceiling limitation inherent in Single-Source RAG. Recent studies (Yan et al., 2024; Yao et al., 2023; Wang et al., 2024; Zhao et al., 2024a) demonstrate that leveraging multiple retrieval sources dynamically and controllably can improve retrieval accuracy, thereby enabling the generation of more comprehensive and high-quality knowledge-grounded responses. For instance, ReAct (Yao et al., 2023) introduces an iterative multi-source retrieval approach, whereas UniMS-RAG (Wang et al., 2024) uniquely integrates source selection, retrieval, and generation into a unified model enhanced with action and evaluation tokens, enabling the language model to dynamically invoke and filter sources based on real-time demands. Additionally, PrefRAG (Zhao et al., 2024a) advances RAG further by employing a preference-driven adaptive retrieval mechanism coupled with self-reflection, supporting in-depth and controlled exploration across diverse retrieval sources. Despite these advances, most existing RAGs treat source-level retrieval in a simplistic manner—either aggregating all sources uniformly or statically selecting a single source—thereby neglecting the semantic diversity and redundancy inherent across sources. Furthermore, even after document retrieval, many systems (e.g., BM25 (Robertson et al.,

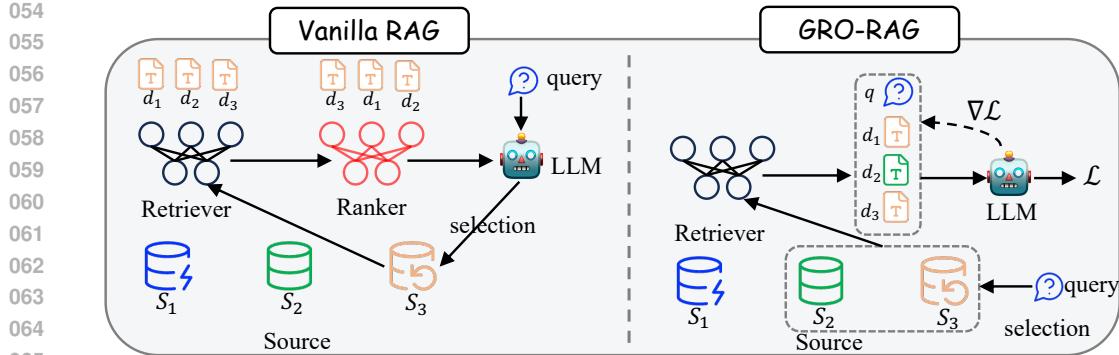


Figure 1: Comparison between vanilla RAG and our proposed GRO-RAG. Vanilla RAG pipelines retrieve documents from certain source and rank them based on query-document relevance. GRO-RAG selects a subset of sources that jointly balance query relevance and semantic diversity. And it re-ranks retrieved documents using a single backward pass over the generation loss, estimating each document’s contribution via the alignment between its representation and the loss gradient.

2009)) rely on re-ranking models that score documents solely based on retrieval-level signals such as query-document similarity, without considering their actual utility to the downstream generation objective. This creates a mismatch between what is retrieved and what is ultimately needed for high-quality generation.

To address these limitations, we propose GRO-RAG, a training-free optimization framework tailored for Multi-source Retrieval-Augmented Generation (MS-RAG) (Yan et al., 2024; Yao et al., 2023; Wang et al., 2024; Zhao et al., 2024a). GRO-RAG introduces a principled mechanism to dynamically select both *“which source combinations to retrieve from”* and *“which documents to use as context”*. Firstly, we select source combinations by optimizing a relevance-redundancy tradeoff that jointly considers query relevance and inter-source semantic overlap, rather than uniformly aggregating or statically selecting sources. This allows GRO-RAG to identify a diverse yet relevant set of sources that provide complementary information, improving both recall and retrieval quality at the source level. Subsequently, we introduce a gradient-aware document re-ranking strategy that directly leverages the feedback from large language model. Specifically, after retrieving candidate documents from the selected sources, we perform a forward pass to compute the generation loss and then a single backward pass to obtain the gradient of this loss with respect to the input representations. For each document, we compute an importance score as the inner product between its hidden representation and the loss gradient. This score reflects how much the document contributes to reducing the generation loss. Selecting the Top-k documents based on these scores enables a posterior-aware re-ranking process that is tightly aligned with the final generation objective, rather than heuristic query-document similarity. GRO-RAG is fully training-free and compatible with frozen language models. It introduces no additional model parameters or training phases, and requires only one forward-backward pass per query. Theoretically, we show that the gradient-based Top-k selection approximates the solution to an underlying utility maximization problem and aligns with minimizing a leave-one-out loss upper bound.

Our main contributions are summarized as follows:

- We introduce a novel, training-free approach that estimates each document’s contribution to the generation loss using a single backward pass, enabling posterior-aware Top-k selection based on model feedback.
- We formulate source selection as a relevance-redundancy tradeoff over source subsets, enabling dynamic source combination and improving retrieval diversity and complementarity.
- We theoretically show that our gradient-based selection approximates an underlying utility maximization objective and aligns with minimizing a leave-one-out loss upper bound. Extensive experiments on multi-source QA and open-domain generation benchmarks demonstrate consistent improvements over strong retrieval and re-ranking baselines.

108 **2 RELATED WORKS**109 **2.1 RETRIEVAL-AUGMENTED GENERATION**

110 Retrieval-augmented generation (RAG) is increasingly recognized as an effective method to mitigate
 111 several limitations of large language models (LLMs), notably hallucinations (Shuster et al., 2021),
 112 factuality issues (Wang et al., 2023), and the lack of long-term memory (Xu et al., 2022). Following
 113 the established "Retriever-and-Reader" paradigm (Chen et al., 2017; Das et al., 2019), RAG first
 114 employs an external retriever to select relevant textual information from knowledge sources (e.g.,
 115 Wikipedia). These retrieved passages subsequently serve as external context for a reader/generator,
 116 enabling the model to generate knowledge grounded response (Lewis et al., 2020). Initial retrieval
 117 methods (e.g., BM25 (Robertson et al., 2009)) used sparse retriever for relevance scoring, but often
 118 fail to capture deeper semantic information (Guo et al., 2022). To overcome this limitation, language-
 119 model-based dense retrieval approaches have been developed, encoding documents and queries
 120 into dense vectors to effectively represent the semantic feature of text content (Karpukhin et al.,
 121 2020; Li et al., 2023; Bruch et al., 2023). Recently, researchers have explored leveraging LLMs as
 122 retrievers (Wang et al., 2024; Asai et al., 2023; Jiang et al., 2023; Yu et al., 2024; Zhao et al., 2024a).
 123 For instance, Self-RAG introduces reflection tokens, enabling the model to dynamically retrieve
 124 supporting passages and self-correct its outputs (Asai et al., 2023).
 125

126 **2.2 MULTI-SOURCE RAG**

127 Based on the retrieval sources, recent advances in retrieval-augmented generation (RAG) can be
 128 categorized into Single-Source RAG (SS-RAG) (Asai et al., 2023; Jiang et al., 2023; Yu et al., 2024)
 129 and Multi-Source RAG (MS-RAG). SS-RAG methods inherently limit RAG system performance
 130 due to reliance on a single knowledge source. MS-RAG addresses this limitation by integrating
 131 heterogeneous knowledge sources, including specialized databases, structured archives, and the open
 132 web (Yan et al., 2024). Common MS-RAG implementations perform sequential or parallel retrieval
 133 across multiple indices. For example, CRAG treats the web as a fallback source (Yan et al., 2024),
 134 while ReAct coordinates retrieval and reasoning within an agent-based framework (Yao et al., 2023).
 135 However, naively concatenating evidence from diverse sources can enlarge context windows and
 136 introduce noise or conflicting information, thus necessitating *adaptive source selection* (Wang et al.,
 137 2024; Zhao et al., 2024a). UniMS-RAG (Wang et al., 2024) addresses these issues by unifying source
 138 selection, retrieval, and generation into a single sequence-to-sequence model. It introduces specific
 139 action and evaluation tokens, allowing the LLM to dynamically invoke and filter sources as needed.
 140 PrefRAG (Zhao et al., 2024a) further advances MS-RAG with a preference-driven adaptive retrieval
 141 approach that employs self-reflection, enabling more in-depth and controllable exploration across
 142 multiple retrieval sources.
 143

144 **3 METHOD**145 **3.1 TASK DEFINITION AND NOTATION**

146 Let \mathcal{Q} be the space of user queries and \mathcal{A}^* the space of target answers. We are given a query $q \in \mathcal{Q}$
 147 and seek to generate an answer a^* with the assistance of external evidence. Evidence is organized in
 148 a set of *heterogeneous sources* $\mathcal{S} = \{s_1, \dots, s_{|\mathcal{S}|}\}$. Each source $s \in \mathcal{S}$ exposes a (possibly dynamic)
 149 document collection $\mathcal{D}_s = \{d_{s,1}, \dots, d_{s,|\mathcal{D}_s|}\}$, where a document d is a sequence of tokens from the
 150 vocabulary \mathcal{V} . Our goal is to choose both (i) a subset of sources $\mathcal{A}^* \subseteq \mathcal{S}$ and (ii) a size- k document
 151 set $D^*(q) \subseteq \bigcup_{s \in \mathcal{A}^*} \mathcal{D}_s$ that minimizes the generation loss. A frozen Transformer (Vaswani et al.,
 152 2017) language model \mathcal{M}_θ receives the query q together with a small context set D^* and produces a
 153 best answer a^* . Because enumerating all subsets is infeasible, the following two subsections describe
 154 a tractable, training-free procedure—first selecting sources, then ranking documents with gradient
 155 feedback—that approximates the optimum of generation objective.
 156

157 **3.2 SOURCE COMBINATION SELECTION**

158 Suppose a geography query asks “Which European river flows through both Vienna and Bratislava?”.
 159 A news feed, Wikipedia, and a travel blog may each contain statements answering the question. If we
 160

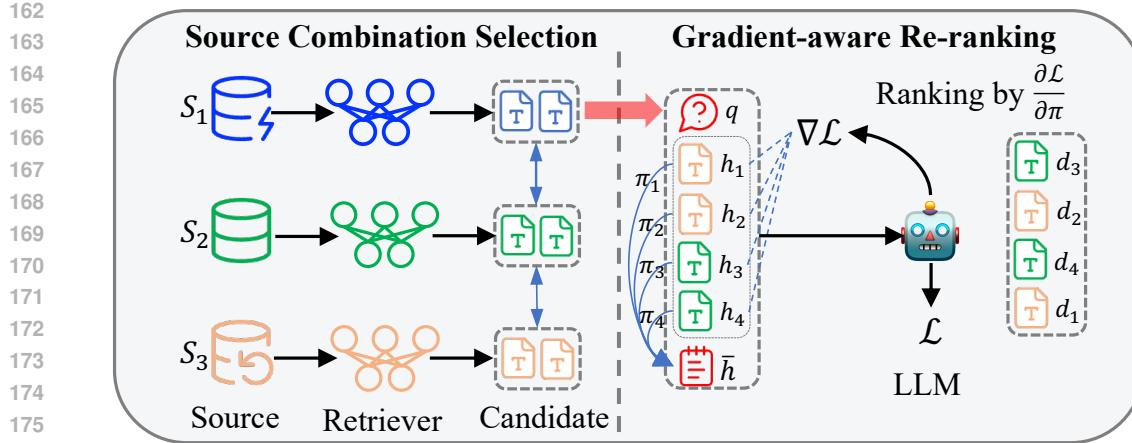


Figure 2: Model architecture of GRO-RAG

indiscriminately merge all three sources the model will read many redundant sentences about the Danube, wasting context budget. Conversely, if we keep only Wikipedia, we indeed cover the Danube, but we throw away the travel-blog anecdote that might help the model answer follow-up questions (e.g. travel time, boat service). Our goal is therefore to retain sources that bring new information while discarding those that merely repeat what has already been covered.

Selecting *which source combinations to retrieve from* is crucial in multi-source RAG (Wang et al., 2024; Zhao et al., 2024a): querying too many sources balloons the candidate pool with near-duplicates, whereas restricting to a single source squanders complementary evidence that may be indispensable for multi-hop reasoning. We formalize this decision as a small yet expressive *relevance–redundancy optimization* and solve it with a greedy algorithm that enjoys a provable approximation ratio.

Our first step is to choose a subset of sources $\mathcal{A} \subseteq \mathcal{S}$ with the given query q . For each source $s \in \mathcal{S}$ we invoke a fixed recall engine¹ and obtain the top- m candidates $\mathcal{C}_s(q) = \{d_{s,1}, \dots, d_{s,m}\}$. $\mathbf{q} = g_{\text{enc}}(q)$ is the query embedding of q . A source representation $\mathbf{s} \in \mathbb{R}^d$ is formed by averaging the frozen document embeddings² of its candidates:

$$\mathbf{s} = \frac{1}{m} \sum_{j=1}^m \mathbf{d}_{s,j}, \quad \mathbf{d}_{s,j} = g_{\text{enc}}(d_{s,j}). \quad (1)$$

We define a scoring function $f : 2^{\mathcal{S}} \rightarrow \mathbb{R}$ over subsets of sources for a candidate subset $\mathcal{A} \subseteq \mathcal{S}$ as:

$$f(\mathcal{A}; \lambda) = \sum_{s \in \mathcal{A}} \cos(\mathbf{q}, \mathbf{s}) - \lambda * \sum_{s, s' \in \mathcal{A}, s < s'} \cos(\mathbf{s}, \mathbf{s}'), \quad (2)$$

where the first term measures query relevance and the second quantifies inter-source redundancy, and $\lambda \in (0, 1)$ trades off relevance and redundancy. The relevance term favors sources containing content aligned with the query semantics; the redundancy term penalizes sources whose content is highly similar to one another. It can be read as reward the marginally informative, punish the already-covered.

We now turn to the sub-modularity of equation 2. The first term is modular by construction, as it is a sum of independent contributions. The redundancy term is pairwise and symmetric, and under cosine similarity, it induces a sub-modular structure due to increasing overlap with more sources. Since the sum of a modular and a sub-modular function remains sub-modular, the full objective f is submodular (Carbonell & Goldstein, 1998) when λ is small. This structure allows us to employ a greedy algorithm for subset selection with a provable approximation guarantee. We initialize $\mathcal{A}_0 = \emptyset$. For $t = 0, \dots, |\mathcal{S}| - 1$ we compute the marginal gain $\Delta_t(s) = f(\mathcal{A}_t \cup \{s\}; \lambda) - f(\mathcal{A}_t; \lambda)$ for every

¹Any sparse or dense retriever is admissible; the choice does not affect downstream optimization.

²We use sentence-BERT; other encoders yield similar behavior.

216 $s \in \mathcal{S} \setminus \mathcal{A}_t$ and set $\mathcal{A}_{t+1} = \mathcal{A}_t \cup \{\arg \max_s \Delta_t(s)\}$. The loop stops once $|\mathcal{A}_{t+1}| = |\mathcal{S}|$ or no positive
217 gain exists.

218 We denote the final set by $\mathcal{A}_{\text{greedy}}$, classical results on submodular (Carbonell & Goldstein, 1998)
219 maximization imply the following guarantee:
220

$$f(\mathcal{A}_{\text{greedy}}; \lambda) \geq (1 - 1/e) \max_{\mathcal{A} \subseteq \mathcal{S}, |\mathcal{A}| \leq |\mathcal{S}|} f(\mathcal{A}; \lambda). \quad (3)$$

223 The union of their candidate sets $\mathcal{C}^*(q) = \bigcup_{s \in \mathcal{A}_{\text{greedy}}} \mathcal{C}_s(q)$ serves as the input to the gradient-aware
224 re-ranking stage. This two-level selection retains complementary evidence while sharply reducing
225 cross-source redundancy.

226 3.3 GRADIENT-AWARE RE-RANKING

228 Our source selection stage has already removed most irrelevant or duplicated corpora, leaving a mixed
229 candidate pool $\mathcal{C}^*(q) = \bigcup_{s \in \mathcal{A}^*} \mathcal{C}_s(q) = \{d_1, \dots, d_n\}$. The final bottleneck is the frozen language
230 model’s context limit: it can absorb at most k passages. The usual remedy is to train a cross-encoder
231 or to reuse query–document similarity scores, but both approaches ignore how the generator itself
232 reacts to each passage. GRO-RAG therefore lets the LLM “vote”: we ask, “If I boost passage i , will
233 my loss decrease?”—and we obtain the answer from a single forward–backward pass.

234 For each candidate document $d_i \in \mathcal{C}^*(q)$ we already possess the frozen contextual embedding \mathbf{h}_i .
235 Selecting exactly k passages is combinatorial. We relax the binary choice (keep / drop) to a soft
236 non-negative weight vector $\pi \in \Delta^n$ ($\pi_i \in [0, 1]$, $\|\pi\|_1 = 1$), we write the mixture representation
237 $\bar{h}(\pi) = \sum_{i=1}^n \pi_i \mathbf{h}_i$. Next, we construct a soft prompt $\langle q, \bar{h}(\pi) \rangle$, and compute the generation loss
238 $\mathcal{L}(\pi) = \mathcal{L}(a^* | q, \bar{h}(\pi))$ with respect to the reference answer a^* . Although this relaxation converts
239 the discrete Top- k selection into a continuous problem, the resulting loss function $\mathcal{L}(\cdot)$ remains
240 non-convex and analytically intractable due to the non-linear behavior of the generator. To obtain
241 a tractable approximation, we apply a first-order Taylor expansion around the uniform mixture
242 $\bar{\pi} = (1/n, \dots, 1/n)$. This yields:

$$\mathcal{L}(\pi) \approx \mathcal{L}(\bar{\pi}) + \langle \nabla_{\bar{h}} \mathcal{L}, \bar{h}(\pi) - \bar{h}(\bar{\pi}) \rangle \quad (4)$$

243 Considering $\bar{h}(\pi) = \sum_{i=1}^n \pi_i \mathbf{h}_i$ and $\bar{h}(\bar{\pi}) = \frac{1}{n} \sum_{i=1}^n \mathbf{h}_i$, we obtain:
244

$$\mathcal{L}(\pi) \approx \mathcal{L}(\bar{\pi}) + \sum_{i=1}^n \pi_i \langle \nabla_{\bar{h}} \mathcal{L}, \mathbf{h}_i \rangle \quad (5)$$

245 Minimizing the loss is therefore approximately equivalent to minimizing a linear weighted sum over
246 document scores. Since π is constrained to lie in a k -sparse simplex, the approximate optimal solution
247 is obtained by selecting the k documents with the largest negative inner products $\langle \mathbf{h}_i, -\nabla_{\bar{h}} \mathcal{L} \rangle$. We
248 define the ranking score of document d_i accordingly:

$$\phi_i = \langle \mathbf{h}_i, -\nabla_{\bar{h}} \mathcal{L} \rangle. \quad (6)$$

249 This value estimates the sensitivity of the generation loss to the presence of d_i in the prompt, and
250 serves as a posterior-aware signal for document selection. The larger ϕ_i , the more sharply the loss
251 would drop if document i received additional weight. We keep the k documents with the highest ϕ_i ,
252 set all other weights to zero, and re-normalize. The selected passages $D^*(q)$ are finally pre-pended to
253 the query. In practice we simply concatenate their raw text, but one could alternatively keep the mixed
254 vector \bar{h} to save context tokens. That is, we only need one forward-backward pass to sort the candidate
255 documents. This strategy requires no additional training, and leverages LLM-internal gradients to
256 estimate document utility with respect to the actual generation objective, not just similarity.

257 A natural way to measure how much a single document d_i actually helps the generator is to remove
258 it from the context, run the model again, and observe how much the loss increases. We call this
259 *leave-one-out* (LOO) loss. Formally, starting from the uniform mixture $\bar{\pi} = (\frac{1}{n}, \dots, \frac{1}{n})$ over all n
260 candidates, we define

$$\mathcal{L}_{\text{loo}}(d_i) = \underbrace{\mathcal{L}(\bar{\pi})}_{\text{all documents}} - \underbrace{\mathcal{L}(\bar{\pi} - \frac{1}{n} \mathbf{e}_i)}_{\text{document } i \text{ removed}}, \quad (7)$$

261 where \mathbf{e}_i is the i -th basis vector. Computing this quantity for every passage would require $n+1$
262 forward passes—prohibitively slow. Below we show that a *single backward pass* yields a gradient
263 score ϕ_i that upper-bounds LOO loss, thereby providing a safe ranking surrogate.

270 **Proposition 3.1 (Gradient inner product upper-bounds leave-one-out)** *Let $g = \nabla_{\bar{h}} \mathcal{L}(\bar{\pi})$ and
 271 $\phi_i = \langle \mathbf{h}_i, -g \rangle$. If the one-dimensional function $\ell_i(t) = \mathcal{L}(\bar{\pi} + t \mathbf{e}_i)$ is convex on the interval
 272 $t \in [-\frac{1}{n}, \varepsilon]$ for some $\varepsilon > 0$, then for every documents d_i*

$$\mathcal{L}_{\text{loo}}(d_i) \leq -\phi_i.$$

275 Thus ϕ_i upper-bounds the true marginal utility, so ranking by ϕ is guaranteed to prioritize passages
 276 whose absence would hurt the loss the most.

277 The gradient-based scoring described above is a single-step forward-backward pass: we linearize the
 278 loss landscape around a uniform mixture $\bar{\pi} = (\frac{1}{n}, \dots, \frac{1}{n})$ and select the top- k documents accordingly.
 279 Moreover, we can extend this process into a multi-step optimization routine. At each iteration t ,
 280 we maintain a soft weight vector π^t , use it to form the context $\bar{h}(\pi^t) = \sum_i \pi_i^t \mathbf{h}_i$, and compute
 281 the generation loss and its gradient via a forward–backward pass. This yields the descent direction
 282 $g^t = \nabla_{\bar{h}} \mathcal{L}(\bar{h}(\pi^t))$, which we use to update π^t via gradient descent. The updated weights are then
 283 projected back onto the k -sparse simplex to maintain feasibility. In effect, this defines an iterative
 284 refinement process over document mixtures, allowing the model itself to guide the selection toward
 285 increasingly informative subsets. Each iteration re-evaluates the loss under the *current* document
 286 mixture and refines the weights accordingly—no new model parameters are introduced, and the only
 287 computation is an additional forward-backward pass of the frozen LLM.

288 **Proposition 3.2 (Linear convergence of the iterative loop)** *Let $\mathcal{L}(\pi) = \mathcal{L}(a^* \mid q, \bar{h}(\pi))$ denote
 289 the generation loss evaluated at $\bar{h}(\pi) = \sum_{i=1}^n \pi_i \mathbf{h}_i$. Assume μ -strong convexity and L -smoothness
 290 of \mathcal{L} in the sub-space $\text{span}\{\mathbf{h}_1, \dots, \mathbf{h}_n\}$. Starting from the uniform vector $\bar{\pi} = (1/n, \dots, 1/n)$,
 291 repeat*

$$\tilde{\pi}^{t+1} = \pi^t - \eta \nabla_{\pi} \mathcal{L}(\pi^t), \quad \pi^{t+1} = \Pi_{\Delta^{n-1}}(\tilde{\pi}^{t+1}), \quad 0 < \eta \leq 1/L,$$

293 where $\Pi_{\Delta^{n-1}}$ projects onto the probability simplex $\Delta^{n-1} = \{\pi \geq 0, \|\pi\|_1 = 1\}$. Then for all $t \geq 0$:

$$\mathcal{L}(\pi^{t+1}) - \mathcal{L}^* \leq (1 - \eta\mu) [\mathcal{L}(\pi^t) - \mathcal{L}^*],$$

295 with optimal $\mathcal{L}^* = \min_{\pi \in \Delta^{n-1}} \mathcal{L}(\pi)$. Hence each additional iteration contracts the sub-optimality
 296 by the factor $(1 - \eta\mu)$ and therefore never worsens the one-step solution. A proof is provided in
 297 Appendix B.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETTINGS

303 **Dataset** Following previous works (Yao et al., 2023; Trivedi et al., 2022a; Zhao et al., 2024b), We
 304 evaluate our method on four widely used question answering benchmarks that span both open-domain
 305 and multi-hop reasoning settings: **HotpotQA** (Yang et al., 2018), **2WikiMultihopQA** (Ho et al.,
 306 2020), and **MuSiQue** (Trivedi et al., 2022b). These datasets each provide a set of ground-truth
 307 documents (typically 10-20) for each question as well as ground-truth answers.

309 **Metrics** We assess model performance from both retrieval and generation perspectives. For retrieval
 310 evaluation, we use **nDCG@ k** (**Normalized Discounted Cumulative Gain**) (Burges et al., 2005),
 311 which measures the ranking quality of selected documents based on graded relevance and position. A
 312 higher nDCG indicates that relevant documents are ranked closer to the top. Ground-truth relevance
 313 annotations—where available—are used as supervision. For generation evaluation, we adopt two
 314 standard QA metrics (Gao et al., 2023): **Exact Match (EM)**, which measures the proportion of
 315 generated answers that match ground-truth strings exactly, and **F1 score**, which captures the overlap
 316 between predicted and reference answers based on precision and recall. Retrieval metrics are
 317 computed over the Top- k ranked documents, while generation metrics are reported over answers
 318 produced by the language model conditioned on the selected document set. Here, we set $k = 10$.

319 **Retrieval settings** To support multi-source retrieval, we leverage both local corpora and web
 320 sources. Specifically, for web search, we employ the DuckDuckGo API—a publicly available
 321 interface—to access large-scale online information. We compare with three representative retrievers:
 322 **BM25** (Robertson et al., 2009), a sparse keyword-based method; **E5-base** (Wang et al., 2022), a dense
 323 dual-encoder trained via contrastive learning; and **BGE-M3** (Chen et al., 2024), a dense multilingual
 and multitask-aligned retriever.

324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
Table 1: Results (%) of GRO-RAG and baselines on three datasets. "Bold" and "Underlined" denote the highest absolute values and second highest values, respectively.

LLM	RAG methods	HotpotQA		2WikimQA		MuSiQue	
		F1	EM	F1	EM	F1	EM
Llama3.1-8B	w/o Retrieval	-	27.8	23.1	19.7	13.9	8.4
	Vanilla RAG	Local	34.2	28.2	24.1	19.4	13.1
	Vanilla RAG	Web	31.5	24.8	20.4	15.3	10.5
	Vanilla RAG	Both	36.0	29.7	27.3	21.8	15.9
	Single-Source RAG	Self-RAG	32.3	26.4	21.1	17.4	14.8
	Single-Source RAG	FLARE	34.5	28.6	28.5	23.0	17.3
	Single-Source RAG	RankRAG	31.9	24.3	25.7	20.8	13.6
	Multi-Source RAG	CRAG	34.2	25.5	22.6	17.9	16.2
	Multi-Source RAG	GRO-RAG	39.1	30.9	28.9	22.8	18.6
	w/o Retrieval	-	29.4	23.6	18.6	13.5	10.3
GLM-4	Vanilla RAG	Local	36.8	29.8	25.3	20.1	13.0
	Vanilla RAG	Web	30.4	23.9	19.5	14.8	9.4
	Vanilla RAG	Both	39.3	31.5	28.2	22.4	16.5
	Single-Source RAG	Self-RAG	34.4	28.3	22.8	18.3	17.8
	Single-Source RAG	FLARE	38.6	30.7	29.7	23.8	20.2
	Single-Source RAG	RankRAG	33.2	27.3	27.4	21.6	15.8
	Multi-Source RAG	CRAG	38.1	30.3	24.8	20.4	17.4
	Multi-Source RAG	GRO-RAG	42.8	33.6	30.3	23.7	21.1
	w/o Retrieval	-	42.8	33.6	30.3	23.7	21.1
	w/o Retrieval	-	42.8	33.6	30.3	23.7	21.1

Table 2: Comparison of NDCG@10 for different re-ranking methods on three QA datasets using only local corpus. GRO-RAG uses gradient-based re-ranking from frozen LLMs (LLaMA3-8B and GLM-4), while BM25, BGE-M3, and E5-base serve as heuristic or dense retrieval baselines.

Different rerankers	HotpotQA	2WikimQA	MuSiQue	Average
BM25	0.6237	0.5760	0.3453	0.5150
BGE-M3	0.6892	0.6273	0.3922	0.5696
E5-base	0.7013	0.6749	0.4180	0.5981
GRO-RAG with Llama3.1-8B	0.6442	0.6345	0.4039	0.5609
GRO-RAG with GLM-4	0.6538	0.6382	0.4156	0.5692

Generation settings We compare GRO-RAG against four categories of baselines. (1) w/o-Retrieval: LLM directly answers questions without access to any external documents. (2) Vanilla RAG: Standard RAG methods that retrieve documents from either a local corpus, a web corpus, or both, and concatenate them with query as input to the LLM. (3) Single-source RAG: Methods such as Self-RAG (Asai et al., 2023), FLARE (Jiang et al., 2023), and RankRAG (Yu et al., 2024), which rely on a single retrieval source (e.g., local corpus only). (4) Multi-source RAG includes CRAG (Yan et al., 2024) conduct one-time retrieval from the primary source, followed by a one-time supplementary retrieval from a secondary source. For all methods, we conduct experiments based on two built-in LLMs, including Llama3.1-8B (Grattafiori et al., 2024; Meta AI, 2024) and GLM4 (GLM et al., 2024).

4.2 MAIN RESULTS

Performance of Generation Table 1 presents a comprehensive comparison across three representative QA benchmarks—HotpotQA, 2WikiMQA, and MuSiQue—under both LLaMA3.1-8B and GLM-4 language models. We begin by observing the general trend that retrieval substantially boosts performance over generation-only settings. Across all datasets and models, methods with access to external documents consistently outperform the No-Retrieval baseline, confirming the necessity of knowledge augmentation for multi-hop reasoning tasks. Among vanilla strategies, retrieving from the local corpus tends to be more effective than the web corpus alone, likely due to better domain alignment and lower noise. Concatenating both further improves accuracy, suggesting local and web sources offer complementary coverage. However, simply merging top documents from multiple

378 Table 3: Ablation study of GRO-RAG. Removing source selection (w/o SCS) or gradient re-ranking
 379 (w/o GR) leads to performance degradation across models and datasets.

381	382	LLM	Methods	HotpotQA		2WikimQA		MuSiQue	
				F1	EM	F1	EM	F1	EM
383	384	Llama3.1-8B	GRO-RAG	39.1	30.9	28.9	22.8	18.6	10.3
			w/o SCS	38.0	30.6	26.4	21.3	17.0	10.2
			w/o GR	37.5	30.2	23.3	19.6	16.2	9.3
385	386	GLM-4	GRO-RAG	42.8	33.6	30.3	23.7	21.1	12.4
			w/o SCS	40.1	31.4	28.6	22.5	20.0	11.5
			w/o GR	37.6	28.7	25.3	20.9	16.8	9.4

387 sources is suboptimal. Vanilla RAG (Both) and CRAG, while better than single-source retrieval,
 388 rely on static combination rules and lack an understanding of which sources are truly informative.
 389 GRO-RAG addresses this through a source combination module that jointly considers query relevance
 390 and semantic redundancy, selecting a subset of sources that are diverse and non-overlapping.
 391 This leads to improved retrieval precision while avoiding wasted context on near-duplicate content.
 392 Compared to adaptive single-source methods such as FLARE and Self-RAG, GRO-RAG consistently
 393 achieves stronger performance across all datasets. These methods typically depend on fallback
 394 heuristics or self-generated queries, which may help in certain cases but do not explicitly model
 395 the final generation objective. In contrast, GRO-RAG performs gradient-based re-ranking directly
 396 using the LLM’s generation loss, enabling the model to “vote” on which passages are most useful
 397 via a single backward pass. This strategy aligns document selection tightly with the downstream
 398 objective and proves especially effective under tight context budgets. On more challenging datasets
 399 such as MuSiQue, which feature higher document entropy and require more subtle reasoning, the
 400 advantage of generation-aware selection becomes even more evident. GRO-RAG not only main-
 401 tains high accuracy but also demonstrates greater stability across LLMs of varying capacities. For
 402 instance, while many baselines suffer performance drops when moving from GLM-4 to the smaller
 403 LLaMA3.1-8B, GRO-RAG’s relative improvements remain consistent, highlighting its robustness
 404 and model-agnostic nature. Lastly, we emphasize that GRO-RAG achieves these gains without any
 405 fine-tuning or additional model parameters. All retrieval and scoring steps are conducted with frozen
 406 models, making GRO-RAG readily deployable in practical settings.

407 **Performance of Retrieval** To ensure fair comparison and isolate the impact of re-ranking, we
 408 constrain retrieval to a single local corpus and use identical candidate pools for all methods in a
 409 single source setting. The results are shown in Tab 2. We compare GRO-RAG against three common
 410 baselines: BM25, a sparse term-matching method; BGE-M3, a modern dense retriever trained
 411 with contrastive supervision; and E5-base, a strong general-purpose embedding model. GRO-RAG
 412 uses a training-free re-ranking approach that computes gradient-based importance scores from the
 413 frozen language model. Despite not using any retrieval supervision or document–query similarity
 414 learning, GRO-RAG consistently improves over BM25 and performs on par with or close to strong
 415 dense retrievers. In particular, on the most difficult dataset (MuSiQue), GRO-RAG achieves higher
 416 nDCG than BGE-M3, suggesting that gradient signals from the generation objective can capture
 417 nuanced relevance signals beyond static embeddings. These results demonstrate that even without
 418 additional training, GRO-RAG can effectively identify useful documents for the LLM, narrowing the
 419 gap to supervised retrievers and providing a principled, efficient alternative for ranking in retrieval-
 420 augmented generation. We emphasize that GRO-RAG is not designed to be a standalone retriever; its
 421 goal is not to maximize retrieval metrics, but to identify documents most useful for generation. In
 422 subsequent experiments, we will show that this strategy leads to consistent improvements in final
 423 answer accuracy with minimal computational overhead.

424 4.3 IN-DEPTH ANALYSIS

425 **Ablation study** To examine the contribution of each component in GRO-RAG, we conduct an
 426 ablation study by removing (1) the Source Combination Selection (SCS) module and (2) the Gradient-
 427 aware Re-ranking (GR) module. The results are summarized in Table 3. Removing SCS means
 428 retrieving documents independently from all sources without optimizing for relevance-redundancy

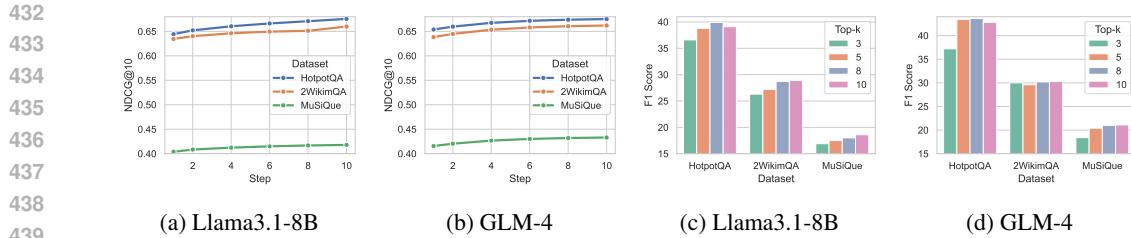


Figure 3: (a)(b) show the effect of optimization steps on retrieval performance (NDCG@10); (c)(d) show the effect of Top-k values on generation performance (F1 score).

tradeoff. While this variant still benefits from having access to multiple corpora, it introduces redundant or semantically overlapping documents into the context, leading to a modest but consistent performance drop across datasets. This suggests that carefully selecting a compact and complementary set of sources helps improve information coverage while minimizing wasteful input. On the other hand, removing GR has a more pronounced effect. This variant replaces gradient-based scoring with a naive per-document loss, ignoring how documents interact when presented jointly. As a result, the model is more likely to prioritize individually informative but contextually redundant documents. Performance declines most noticeably on complex datasets like MuSiQue, indicating that gradient-informed scores better reflect generation utility under constrained context windows.

Multi-step optimization Figures 3a and 3b illustrate the NDCG@10 of GRO-RAG under multi-step optimization on three QA datasets using LLaMA3.1 and GLM-4 as backbone LLMs, respectively. In both settings, the curves show consistent and steady improvements with an increasing number of optimization steps. Importantly, the gain in each step diminishes at a near-constant rate, forming a clear linear improvement pattern. This trend empirically supports the theoretical expectation of linear convergence, where the optimization error decreases proportionally across iterations. The convergence behavior is stable across datasets and model scales, with the total gain around (3-5%), validating the efficiency and robustness of GRO-RAG’s gradient-based refinement process.

Impact of different k We investigate how varying the number of retrieved documents (k) affects the final answer quality. As shown in Figures 3c and 3d, increasing k does not always lead to better performance. While a larger k provides more information, it may also introduce irrelevant or noisy content, which can interfere with the generation process. We observe that F1 scores generally improve from $k = 3$ to $k = 5$, but further increases sometimes yield diminishing or even negative returns. This highlights importance of carefully selecting k to balance completeness and relevance in RAG.

Computation times Computing a single-step gradient using LLaMA3.1-8B takes 882ms on average, while dense retrievers like BGE-M3 complete retrieval in just 39ms. Compared to traditional retrievers, gradient-based retrieval with GRO-RAG introduces significantly higher latency—typically 1 to 2 orders of magnitude slower. This motivates our design choice of using GRO-RAG as a re-ranker, where the initial retrieval already provides a strong prior. In such cases, one gradient step is needed to refine the document selection, making the additional computation cost manageable while still improving answer quality.

5 CONCLUSION

In this paper, we introduced GRO-RAG, a training-free and gradient-aware framework for document re-ranking in multi-source Retrieval-Augmented Generation. By directly leveraging generation loss gradients, GRO-RAG identifies Top- k documents that most effectively contribute to the model’s output, moving beyond traditional query-based relevance scoring. At the source level, our approach selects complementary sources by jointly modeling query relevance and inter-source redundancy. Theoretically, we show that our re-ranking objective approximates the solution to a loss-minimizing subset selection problem and aligns with minimizing a leave-one-out upper bound.

486 REFERENCES
487

488 Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to
489 retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on*
490 *Learning Representations*, 2023.

491 Sebastian Bruch, Siyu Gai, and Amir Ingber. An analysis of fusion functions for hybrid retrieval.
492 *ACM Transactions on Information Systems*, 42(1):1–35, 2023. doi: 10.1145/3596512.

493

494 Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg
495 Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd international*
496 *conference on Machine learning*, pp. 89–96, 2005.

497 Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering
498 documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR*
499 *Conference on Research and Development in Information Retrieval*, pp. 335–336. ACM, 1998.
500 doi: 10.1145/290941.291025.

501

502 Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading wikipedia to answer open-
503 domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational*
504 *Linguistics (Volume 1: Long Papers)*, pp. 1870–1879, Vancouver, Canada, 2017. Association for
505 Computational Linguistics.

506 Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. Bge m3-embedding:
507 Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge
508 distillation. *arXiv preprint arXiv:2402.03216*, 2024.

509

510 Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, and Andrew McCallum. Multi-step retriever-
511 reader interaction for scalable open-domain question answering. In *7th International Conference*
512 *on Learning Representations (ICLR)*, New Orleans, LA, USA, 2019.

513 Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. Wizard of
514 wikipedia: Knowledge-powered conversational agents. In *Proceedings of the 7th International*
515 *Conference on Learning Representations (ICLR)*, 2019.

516

517 Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun,
518 Haofen Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A
519 survey. *arXiv preprint arXiv:2312.10997*, 2:1, 2023.

520

521 Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas,
522 Guanyu Feng, Hanlin Zhao, et al. Chatglm: A family of large language models from glm-130b to
523 glm-4 all tools. *arXiv preprint arXiv:2406.12793*, 2024.

524

525 Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad
526 Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of
527 models. *arXiv preprint arXiv:2407.21783*, 2024.

528

529 Jiafeng Guo, Yinqiong Cai, Yixing Fan, Fei Sun, Ruqing Zhang, and Xueqi Cheng. Semantic models
530 for the first-stage retrieval: A comprehensive review. *ACM Transactions on Information Systems*,
40(4):1–42, 2022. doi: 10.1145/3486250.

531

532 Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop
533 qa dataset for comprehensive evaluation of reasoning steps. *arXiv preprint arXiv:2011.01060*,
2020.

534

535 Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang,
536 Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In *Proceedings of the*
537 *2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7969–7992, 2023.

538

539 Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi
Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *EMNLP*
(1), pp. 6769–6781, 2020.

540 Mojtaba Komeili, Kurt Shuster, and Jason Weston. Internet-augmented dialogue generation. In
 541 *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume*
 542 *1: Long Papers)*, pp. 8460–8478, Dublin, Ireland, 2022. Association for Computational Linguistics.
 543

544 Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal,
 545 Heinrich Kütller, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented genera-
 546 tion for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:
 547 9459–9474, 2020.

548 Hang Li, Ahmed Mourad, Shengyao Zhuang, Bevan Koopman, and Guido Zuccon. Pseudo rele-
 549 vance feedback with deep language models and dense retrievers: Successes and pitfalls. *ACM*
 550 *Transactions on Information Systems*, 41(3):62:1–62:40, 2023. doi: 10.1145/3570724.

551 Meta AI. Llama 3 model card, 2024.

552

553 Stephen Robertson, Hugo Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond.
 554 *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009.

555

556 Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. Retrieval augmentation
 557 reduces hallucination in conversation. In *Findings of the Association for Computational Linguis-
 558 tics: EMNLP 2021*, pp. 3784–3803, Punta Cana, Dominican Republic, 2021. Association for
 559 Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.320.

560 Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval
 561 with chain-of-thought reasoning for knowledge-intensive multi-step questions. *arXiv preprint*
 562 *arXiv:2212.10509*, 2022a.

563 Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop
 564 questions via single-hop question composition. *Transactions of the Association for Computational*
 565 *Linguistics*, 10:539–554, 2022b.

566

567 Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez,
 568 Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information*
 569 *Processing Systems*, volume 30, pp. 5998–6008. Curran Associates, Inc., 2017.

570 Cunxiang Wang, Xiaoze Liu, Yuanhao Yue, Xiangru Tang, Tianhang Zhang, Cheng Jiayang, Yunzhi
 571 Yao, Wenyang Gao, Xuming Hu, Zehan Qi, Yidong Wang, Linyi Yang, Jindong Wang, Xing Xie,
 572 Zheng Zhang, and Yue Zhang. Survey on factuality in large language models: Knowledge, retrieval
 573 and domain-specificity, 2023.

574

575 Hongru Wang, Wenyu Huang, Yang Deng, Rui Wang, Zezhong Wang, Yufei Wang, Fei Mi, Jeff Z
 576 Pan, and Kam-Fai Wong. Unims-rag: A unified multi-source retrieval-augmented generation for
 577 personalized dialogue systems. *arXiv preprint arXiv:2401.13256*, 2024.

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

581

582 Xinchao Xu, Zhibin Gou, Wenquan Wu, Zheng-Yu Niu, Hua Wu, Haifeng Wang, and Shihang Wang.
 583 Long time no see! open-domain conversation with long-term persona memory. In *Findings of*
 584 *the Association for Computational Linguistics: ACL 2022*, pp. 2639–2650, Dublin, Ireland, 2022.
 585 Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.207.

586

587 Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. Corrective retrieval augmented generation,
 588 2024. URL <https://arxiv.org/abs/2401.15884>.

589

590 Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov,
 591 and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question
 592 answering. *arXiv preprint arXiv:1809.09600*, 2018.

593

594 Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao.
 595 React: Synergizing reasoning and acting in language models. In *International Conference on*
 596 *Learning Representations (ICLR)*, 2023.

594 Yue Yu, Wei Ping, Zihan Liu, Boxin Wang, Jiaxuan You, Chao Zhang, Mohammad Shoeybi, and
 595 Bryan Catanzaro. Rankrag: Unifying context ranking with retrieval-augmented generation in llms.
 596 *Advances in Neural Information Processing Systems*, 37:121156–121184, 2024.

597
 598 Qingfei Zhao, Ruobing Wang, Yukuo Cen, Daren Zha, Shicheng Tan, and Jie Tang. Prefrag:
 599 Preference-driven multi-source retrieval-augmented generation. *arXiv preprint arXiv:2411.00689*,
 600 2024a.

601
 602 Qingfei Zhao, Ruobing Wang, Xin Wang, Daren Zha, and Nan Mu. Towards multi-source retrieval-
 603 augmented generation via synergizing reasoning and preference-driven retrieval. *arXiv preprint*
 604 *arXiv:2411.00689*, 2024b.

605 A PROOF FOR PROPOSITION 3.1

606
 607 **Proposition A.1 (Gradient inner product upper-bounds leave-one-out)** *Let $g = \nabla_{\bar{h}}\mathcal{L}(\bar{\pi})$ and
 608 $\phi_i = \langle \mathbf{h}_i, -g \rangle$. If the one-dimensional function $\ell_i(t) = \mathcal{L}(\bar{\pi} + t \mathbf{e}_i)$ is convex on the interval
 609 $t \in [-\frac{1}{n}, \varepsilon]$ for some $\varepsilon > 0$, then for every documents d_i*

$$610 \mathcal{L}_{\text{loo}}(d_i) \leq -\phi_i.$$

611 Express LOO as a directional difference. From the definition of $\bar{h}(\pi) = \sum_j \pi_j \mathbf{h}_j$:

$$612 \bar{h}\left(\bar{\pi} - \frac{1}{n} \mathbf{e}_i\right) = \bar{h}(\bar{\pi}) - \frac{1}{n} \mathbf{h}_i.$$

613 Hence $\mathcal{L}_{\text{loo}}(d_i) = \ell_i(0) - \ell_i(-\frac{1}{n})$.

614 Apply convexity of ℓ_i . For a convex function f , $f(y) \geq f(x) + f'(x)(y-x)$. Setting $x = 0$, $y = -\frac{1}{n}$
 615 gives

$$616 \ell_i\left(-\frac{1}{n}\right) \geq \ell_i(0) + \ell'_i(0)\left(-\frac{1}{n}\right).$$

617 Rearrange. Subtract the right-hand side from $\ell_i(0)$:

$$618 \ell_i(0) - \ell_i\left(-\frac{1}{n}\right) \leq \frac{1}{n} \ell'_i(0).$$

619 Convert the directional derivative. Chain rule yields $\ell'_i(0) = \langle \mathbf{h}_i, \nabla_{\bar{h}}\mathcal{L}(\bar{\pi}) \rangle = \langle \mathbf{h}_i, g \rangle$.

620 Substitute and flip sign. Therefore

$$621 \mathcal{L}_{\text{loo}}(d_i) \leq \frac{1}{n} \langle \mathbf{h}_i, g \rangle \leq -\phi_i.$$

622 The factor $\frac{1}{n}$ cancels because it is positive and identical for all passages, leaving the desired inequality.

623 B PROOF FOR PROPOSITION 3.2

624
 625 **Proposition B.1 (Linear convergence of the iterative loop)** *Let $\mathcal{L}(\pi) = \mathcal{L}(a^* \mid q, \bar{h}(\pi))$ denote
 626 the generation loss evaluated at $\bar{h}(\pi) = \sum_{i=1}^n \pi_i \mathbf{h}_i$. Assume μ -strong convexity and L -smoothness
 627 of \mathcal{L} in the sub-space $\mathcal{H} = \text{span}\{\mathbf{h}_1, \dots, \mathbf{h}_n\}$. Starting from the uniform vector $\bar{\pi} = (1/n, \dots, 1/n)$,
 628 repeat*

$$629 \tilde{\pi}^{t+1} = \pi^t - \eta \nabla_{\pi} \mathcal{L}(\pi^t), \quad \pi^{t+1} = \Pi_{\Delta^{n-1}}(\tilde{\pi}^{t+1}), \quad 0 < \eta \leq 1/L,$$

630 where $\Pi_{\Delta^{n-1}}$ projects onto the probability simplex $\Delta^{n-1} = \{\pi \geq 0, \|\pi\|_1 = 1\}$. Then for all $t \geq 0$:

$$631 \mathcal{L}(\pi^{t+1}) - \mathcal{L}^* \leq (1 - \eta\mu) [\mathcal{L}(\pi^t) - \mathcal{L}^*],$$

632 with optimal $\mathcal{L}^* = \min_{\pi \in \Delta^{n-1}} \mathcal{L}(\pi)$. Hence each additional iteration contracts the sub-optimality
 633 by the factor $(1 - \eta\mu)$ and therefore never worsens the one-step solution.

634 **Proof:**

635 We have:

$$636 \Delta^{n-1} = \{\pi \in \mathbb{R}^n \mid \pi_i \geq 0, \|\pi\|_1 = 1\}, \quad \Pi_{\Delta^{n-1}}(v) = \arg \min_{\pi \in \Delta^{n-1}} \|\pi - v\|_2. \quad (8)$$

648 For any iterate π^t define the mixed-context hidden vector $\bar{h}(\pi^t) = \sum_i \pi_i^t \mathbf{h}_i$ and let $G^t = \nabla_{\pi} \mathcal{L}(\pi^t)$.
 649 Because $\bar{h}(\pi) = M\pi$ with $M = [\mathbf{h}_1, \dots, \mathbf{h}_n]$, the chain rule gives
 650

$$651 \quad G^t = M^T \nabla_{\bar{h}} \mathcal{L}(\pi^t). \quad (9)$$

652 \mathcal{L} is L -smooth on \mathcal{H} , hence for any $\pi, v \in \mathbb{R}^n$:

$$653 \quad \mathcal{L}(v) \leq \mathcal{L}(\pi) + \langle \nabla_{\pi} \mathcal{L}(\pi), v - \pi \rangle + \frac{L}{2} \|v - \pi\|_2^2. \quad (10)$$

654 Apply equation 10 with $\pi = \pi^t$ and $v = \tilde{\pi}^{t+1} = \pi^t - \eta G^t$:

$$655 \quad \mathcal{L}(\tilde{\pi}^{t+1}) \leq \mathcal{L}(\pi^t) - \eta \langle G^t, G^t \rangle + \frac{L\eta^2}{2} \|G^t\|_2^2 = \mathcal{L}(\pi^t) - \eta \left(1 - \frac{L\eta}{2}\right) \|G^t\|_2^2. \quad (11)$$

656 Let π^* be a minimizer of \mathcal{L} on Δ^{n-1} . Using the Pythagorean property of Euclidean projection,

$$657 \quad \|\pi^{t+1} - \pi^*\|_2^2 \leq \|\tilde{\pi}^{t+1} - \pi^*\|_2^2. \quad (12)$$

658 Because \mathcal{L} is μ -strongly convex,

$$659 \quad \mathcal{L}(\pi) - \mathcal{L}(\pi^*) \geq \frac{\mu}{2} \|\pi - \pi^*\|_2^2, \quad \forall \pi \in \Delta^{n-1}. \quad (13)$$

660 Combine equation 12 and equation 13 to get

$$661 \quad \mathcal{L}(\pi^{t+1}) - \mathcal{L}(\pi^*) \leq \mathcal{L}(\tilde{\pi}^{t+1}) - \mathcal{L}(\pi^*). \quad (14)$$

662 Another consequence of strong convexity is

$$663 \quad \|G^t\|_2^2 \geq 2\mu [\mathcal{L}(\pi^t) - \mathcal{L}(\pi^*)]. \quad (15)$$

664 Insert equation 15 into the descent lemma equation 11 and use $0 < \eta \leq 1/L \Rightarrow 1 - \frac{L\eta}{2} \geq \frac{1}{2}$:

$$665 \quad \mathcal{L}(\tilde{\pi}^{t+1}) \leq \mathcal{L}(\pi^t) - \eta\mu [\mathcal{L}(\pi^t) - \mathcal{L}(\pi^*)]. \quad (16)$$

666 Subtract $\mathcal{L}(\pi^*)$ from both sides and combine with equation 14:

$$667 \quad \mathcal{L}(\pi^{t+1}) - \mathcal{L}(\pi^*) \leq (1 - \eta\mu) [\mathcal{L}(\pi^t) - \mathcal{L}(\pi^*)]. \quad (17)$$

668 This proves the claimed inequality with $\mathcal{L}^* = \mathcal{L}(\pi^*)$. Iterating the contraction yields

$$669 \quad \mathcal{L}(\pi^t) - \mathcal{L}^* \leq (1 - \eta\mu)^t [\mathcal{L}(\pi^t) - \mathcal{L}^*], \quad (18)$$

670 which is geometric linear convergence with rate $1 - \eta\mu$ (< 1).

671 C LIMITATIONS

672 While GRO-RAG offers a lightweight and training-free alternative for improving document selection
 673 in multi-source RAG, it also presents several limitations. First, the gradient-based re-ranking
 674 procedure requires a backward pass through the frozen LLM, which, although computationally
 675 modest compared to fine-tuning, still introduces latency compared to purely retrieval-side methods.
 676 Second, GRO-RAG’s ranking relies on local linear approximations of the generation loss, which may
 677 become inaccurate when the true loss landscape is highly non-linear or when documents exhibit strong
 678 interaction effects. Finally, our source selection module is limited to fixed document embeddings;
 679 future extensions could explore query-aware or adaptive source encoders to further enhance selection.

680 D BROADER IMPACTS

681 GRO-RAG contributes to the growing field of retrieval-augmented generation by proposing a more
 682 interpretable, modular, and training-free alternative to dense scoring models. Its emphasis on gradient-
 683 based document utility aligns document selection directly with the language model’s generation
 684 behavior, offering greater transparency and controllability in the retrieval pipeline. By avoiding
 685 task-specific fine-tuning, GRO-RAG lowers the resource barriers to deploying RAG systems in
 686 practical applications, particularly in low-resource or privacy-sensitive settings.