WHAT MAKES RETRIEVAL OPTIMAL FOR CONTEXT-LEVEL RAG? A JOINT TOP-k APPROACH

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models can answer user question effectively, but they still suffer from hallucinations and outdated knowledge. Retrieval-Augmented Generation (RAG) has emerged as a popular solution by retrieving accurate and up-to-date reference documents from external databases and incorporating them into the generation context. Most existing theoretical formulations of RAG focus on the document-level setting, where candidate answers are generated for each retrieved document and then ranked. In practice, however, the more commonly deployed paradigm is context-level RAG, which directly generates a final answer conditioned on the top-k retrieved documents. This creates a notable gap between existing theory and practical systems. To bridge this gap, we analyze context-level RAG through a probabilistic marginalization framework and derive a joint top-k retrieval objective whose ranking criterion depends on the question, answer, and comprehensive context. We show that this joint top-k objective is optimal for downstream answer generation. However, directly retrieving according to this criterion is infeasible in practice, since the answer and comprehensive context are unknown at retrieval time. To address this challenge, we propose JBR-C (Joint-Based Retriever for Context-level RAG), a practical retriever that uses listwise learning-to-rank to approximate the latent joint ranking signal with respect to the best answer and context. Extensive experiments on four question-answering benchmarks show that JBR-C consistently outperforms representative retrievers.

1 INTRODUCTION

Large language models (LLMs) have achieved remarkable success and play a central role in a wide range of applications (Kumar, 2024). Despite their strong generative capabilities, LLMs often produce incorrect or fabricated content, a phenomenon commonly referred to as hallucination (Huang et al., 2025a). This issue is particularly pronounced when user queries require knowledge beyond the model's parametric memory or demand up-to-date external information. RAG addresses this limitation by retrieving relevant documents from an external knowledge base and conditioning generation on the retrieved evidence. By grounding responses in external information, RAG has proven effective in reducing hallucinations and improving factual accuracy (Zhu et al., 2025).

Motivated by the practical importance of RAG, substantial research has focused on improving different stages of the pipeline, including chunking (Borgeaud et al., 2022; Yu et al., 2025; Zhong et al., 2025b), embedding (Karpukhin et al., 2020; Shi et al., 2024; Izacard et al., 2022), indexing (Douze et al., 2025; Huang et al., 2025b; Chen et al., 2025; Qiu & Tang, 2025; Zhao et al., 2023), caching (Zhong et al., 2025b; Agarwal et al., 2025; Zhang et al., 2025; Zhong et al., 2025a), and reranking (Chuang et al., 2023; Glass et al., 2022; Catanzaro et al., 2024). Among these components, retrieval quality is especially critical, since the retrieved documents determine the evidence available to the generator. Most existing RAG systems adopt a relevance-based retrieval objective, ranking documents $\{\boldsymbol{d}_i\}$ according to the conditional probability $\Pr(\boldsymbol{d}_i \mid \boldsymbol{q})$ given a question $\boldsymbol{q}$. However, this objective is not always aligned with the downstream goal of maximizing answer quality. A document that appears highly relevant to the query may still provide limited support for generating the correct answer, while a less obviously relevant document may contain decisive evidence. In addition, the common theoretical foundation of RAG is largely derived from the *document-level* formulation (Lewis et al., 2020; Guu et al., 2020), where candidate answers are generated independently for retrieved documents and then reranked. In practice, however, the more widely deployed

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

setting is *context-level* RAG, where the generator directly produces a final answer conditioned on the concatenation of top-k retrieved documents. This creates two important misalignments: (1) between relevance-based retrieval and answer-oriented generation, and (2) between classic RAG theory and context-level RAG practice.

To address these gaps, we analyze context-level RAG through a probabilistic marginalization framework and propose **JBR-C** (Joint-Based Retriever for Context-level RAG), a practical retriever that approximates the latent joint ranking signal using listwise learning-to-rank. Instead of optimizing retrieval solely with query-document relevance, JBR-C learns to rank documents according to their joint utility for generating the best answer. In summary, we make the following contributions:

- Theoretically, we develop a probabilistic framework for context-level RAG and derive a joint top-k retrieval objective whose ranking criterion depends on the question, answer, and comprehensive context. We further prove that this objective is optimal for downstream answer generation.
- We propose JBR-C, a practical retriever that uses listwise learning-to-rank to approximate the latent joint ranking signal. During training, documents are jointly ranked using signals from the question, best answer, and comprehensive context, and the retriever is optimized with a listwise likelihood objective. This design avoids explicit answer probability computation during inference while preserving the benefits of the joint objective.
- Extensive experiments on four question-answering datasets demonstrate that JBR-C consistently outperforms representative dense and sparse retrievers across tasks.

2 Related Work

RAG is typically decomposed into retrieval and generation stages (Fan et al., 2024; Gao et al., 2024). This work focuses on retrieval, which largely determines the quality of downstream generation. Modern RAG systems often incorporate auxiliary components such as document chunking (Borgeaud et al., 2022; Yu et al., 2025; Zhong et al., 2025b), approximate nearest neighbor search (Douze et al., 2025; Huang et al., 2025b; Chen et al., 2025; Qiu & Tang, 2025; Zhao et al., 2023; Wang et al., 2022), caching (Zhong et al., 2025b; Agarwal et al., 2025; Zhang et al., 2025; Zhong et al., 2025a), and reranking (Chuang et al., 2023; Glass et al., 2022; Catanzaro et al., 2024). Despite these, retrieval performance fundamentally depends on the underlying retriever model. Existing retrievers can be broadly categorized into sparse and dense retrievers.

Sparse retrievers. Sparse retrievers rely on lexical matching based on term statistics. Representative methods include TF–IDF (Robertson, 2004) and BM25 (Robertson & Zaragoza, 2009), which remain widely used in RAG systems and hybrid retrieval pipelines (Chuang et al., 2023; Glass et al., 2022; Izacard et al., 2022; Karpukhin et al., 2020; Lee et al., 2019; Mao et al., 2021; Ram et al., 2022; Wang et al., 2023). While effective for exact matching, sparse retrievers often struggle with semantic similarity (e.g.,synonyms). Recent approaches such as SPLADE (Formal et al., 2021b;a; Lassance et al., 2024) improve semantic generalization through learned sparse representations.

Dense retrievers. Dense retrievers encode queries and documents into continuous representations for semantic retrieval. DPR (Karpukhin et al., 2020) popularized the bi-encoder framework and has been adopted in many downstream RAG systems (Lewis et al., 2020; Izacard & Grave, 2021; Glass et al., 2022; Li et al., 2025). Contriever (Izacard et al., 2022) further enables unsupervised dense retrieval through contrastive pretraining on unlabeled corpora. Most dense retrievers optimize query–document relevance without explicitly modeling downstream answer generation. Recent LM-supervised retrievers instead incorporate answer signals during training to better align retrieval with generation. PROMPTAGATOR (Dai et al., 2023) synthesizes task-specific supervision using LLM prompting, while REPLUG (Shi et al., 2024) and RA-DIT (Lin et al., 2023) align retriever distributions with generation probabilities.

Despite recent progress, aligning retrieval with downstream generation remains challenging. Retriever models are typically much smaller than generator LLMs, making direct approximation of generation distributions difficult. Moreover, most prior approaches are motivated by document-level RAG, whereas practical systems predominantly adopt context-level RAG. In contrast, we revisit the probabilistic foundations of context-level RAG and derive a joint ranking objective that unifies retrieval relevance and generation utility. We theoretically show that joint-based retrieval is optimal under the context-level RAG framework, and further develop a practical listwise learning-to-rank

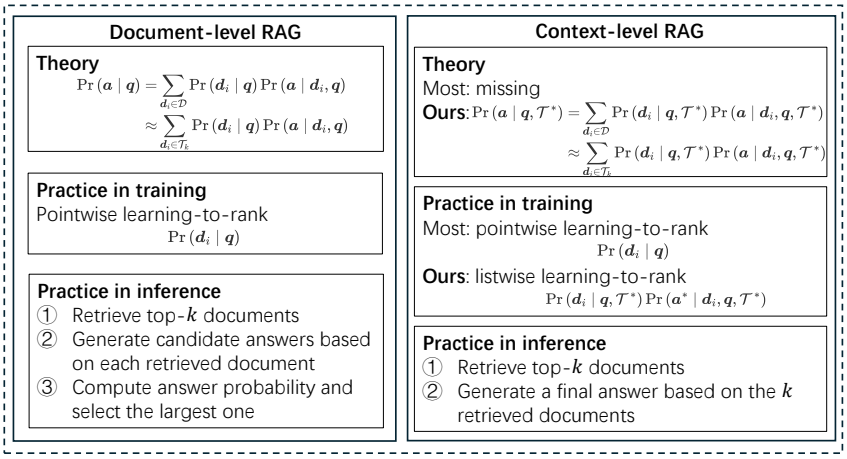


Figure 1: Overview of theoretical framework and retrieval objectives. $\boldsymbol{q}$: question. $\boldsymbol{a}$: answer. $\boldsymbol{d}_i$: document i in the document sets $\mathcal{D}$. $\mathcal{T}_k \subseteq \mathcal{D}$: Top-$k$ retrieval. $\boldsymbol{a}^*$, $\mathcal{T}^*$: best answer and comprehensive context which we defined in Section 3.1.

retriever that achieves strong generation alignment without explicitly computing joint signal during inference. Figure 1 summarizes the differences between prior retrieval paradigms and our approach.

3 Joint Top-k for Context-Level RAG

The document-level theoretical foundation of RAG is from prior work (Lewis et al., 2020; Guu et al., 2020), which models the probability of an answer $\boldsymbol{a}$ given a question $\boldsymbol{q}$ through latent document marginalization. Specifically,

$$\Pr(\boldsymbol{a} \mid \boldsymbol{q}) = \sum_{\boldsymbol{d} \in \mathcal{D}} \Pr(\boldsymbol{d} \mid \boldsymbol{q}) \Pr(\boldsymbol{a} \mid \boldsymbol{d}, \boldsymbol{q}) \approx \tilde{p}(\boldsymbol{a} \mid \boldsymbol{q}; \mathcal{T}_k) = \sum_{\boldsymbol{d} \in \mathcal{T}_k} \Pr(\boldsymbol{d} \mid \boldsymbol{q}) \Pr(\boldsymbol{a} \mid \boldsymbol{d}, \boldsymbol{q}) \quad (1)$$

where $\mathcal{D}$ denotes the full document corpus and $\mathcal{T}_k$ is the retrieved top-k set. When applied during inference, this framework typically requires three steps: ① retrieve the top-k documents, ② generate candidate answers conditioned on each retrieved document, and ③ compute the answer probabilities in Equation 1 to select the best candidate. We refer to this paradigm as *document-level RAG*. However, Steps ② and ③ are computationally expensive, especially when k is large and the question is long. As a result, practical systems often adopt a more efficient alternative: directly generating a final answer conditioned on the retrieved top-k documents . We refer to this setting as *context-level RAG*. Despite its widespread adoption, a principled theoretical foundation for context-level RAG remains underexplored. Moreover, most existing retrieval methods rank documents according to the relevance term $\Pr(\boldsymbol{d} \mid \boldsymbol{q})$. As we later show, this objective is generally suboptimal for context-level RAG, since relevance alone does not necessarily reflect a document set's utility for downstream answer generation. Figure 1 summarizes the relationship between prior retrieval objectives, our theoretical analysis, and the proposed practical method. In Section 3.1, we formulate context-level RAG under a probabilistic marginalization framework. In Section 3.2, we derive a joint-based top-k retrieval objective and prove its optimality under this framework.

3.1 Marginalization Theory in Context-Level RAG

The key to developing a marginalization theory for context-level RAG is to explicitly model interactions among retrieved documents, consistent with practical RAG systems where multiple documents are concatenated into a single context and provided to the generator. Let $\mathcal{D}$ denote the document corpus. We assume an ideal generator that can compute the conditional answer probability given a question $\boldsymbol{q}$ and a retrieval context $\mathcal{T} \subseteq \mathcal{D}$, and then generate the answer with the highest probability, i.e., $\arg\max_{\boldsymbol{a}} \Pr(\boldsymbol{a} \mid \boldsymbol{q}, \mathcal{T})$. We further assume the existence of an oracle best answer $\boldsymbol{a}^*$ for question $\boldsymbol{q}$. Based on these assumptions, we define the notion of a *comprehensive context*.

Definition 1 (Comprehensive Context)**.** *A comprehensive context is a subset of the document corpus that enables an ideal generator to generate the best answer $\boldsymbol{a}^*$ for a question $\boldsymbol{q}$. Formally, let*

$\mathcal{T}^* \subseteq \mathcal{D}$ *denote a document subset such that*

$$\Pr(\boldsymbol{a}^* \mid \boldsymbol{q}, \mathcal{T}^*) \geq \Pr(\boldsymbol{a} \mid \boldsymbol{q}, \mathcal{T}), \quad \forall \boldsymbol{a}, \mathcal{T} \tag{2}$$

A high-quality top-k retrieval result $\mathcal{T}_k \subseteq \mathcal{D}$ should therefore satisfy $\Pr(\boldsymbol{a}^* \mid \boldsymbol{q}, \mathcal{T}_k) \geq \Pr(\boldsymbol{a} \mid \boldsymbol{q}, \mathcal{T}_k), \forall \boldsymbol{a}$. In particular, when $\mathcal{T}_k = \mathcal{T}^*$, the best answer $\boldsymbol{a}^*$ can be generated correctly. However, naively identifying an high-quality context requires evaluating $\Pr(\boldsymbol{a} \mid \boldsymbol{q}, \mathcal{T}_k), \forall \boldsymbol{a}, \mathcal{T}_k \subseteq \mathcal{D}$ which involves $\binom{|\mathcal{D}|}{k}$ possible document combinations for each question–answer pair. This combinatorial search becomes computationally prohibitive when $|\mathcal{D}|$ is large. To reduce the complexity, we derive a decomposed form in which documents can be scored individually for the selection of top-k. Following the marginalization principle used in document-level RAG, we write

$$\Pr(\boldsymbol{a} \mid \boldsymbol{q}, \mathcal{T}^*) = \sum_{\boldsymbol{d} \in \mathcal{D}} \Pr(\boldsymbol{d} \mid \boldsymbol{q}, \mathcal{T}^*) \Pr(\boldsymbol{a} \mid \boldsymbol{d}, \boldsymbol{q}, \mathcal{T}^*) \approx \tilde{p}(\boldsymbol{a} \mid \boldsymbol{q}, \mathcal{T}^*; \mathcal{T}_k) = \sum_{\boldsymbol{d} \in \mathcal{T}_k} \Pr(\boldsymbol{d} \mid \boldsymbol{q}, \mathcal{T}^*) \Pr(\boldsymbol{a} \mid \boldsymbol{d}, \boldsymbol{q}, \mathcal{T}^*) \tag{3}$$

This decomposition reveals three interesting properties. (i) If $\Pr(\boldsymbol{d} \mid \boldsymbol{q}, \mathcal{T}^*)$ is used as a ranking signal, then retrieval can be reduced to scoring individual documents, lowering the search complexity to $O(|\mathcal{D}|)$ when $\mathcal{T}^*$ is known or can be induced by $\boldsymbol{q}$. (ii) Each document $\boldsymbol{d}$ contributes to answer generation in the presence of the comprehensive context $\mathcal{T}^*$, which may contain other complementary supporting documents. This differs fundamentally from document-level RAG, where each document contributes more independently to answer generation and candidate scoring. (iii) The retrieval process implicitly approximates $\Pr(\boldsymbol{a} \mid \boldsymbol{q}, \mathcal{T}^*)$. When $\mathcal{T}_k = \mathcal{T}^*$, the top-$k$ approximation becomes exact. In this case, an ideal generator conditioned on $\boldsymbol{q}$ and $\mathcal{T}_k$ can directly generate $\boldsymbol{a}^*$, without requiring multiple candidate generations or answer ranking as in document-level RAG.

3.2 Optimal Top-k in Context-Level RAG

Based on the marginalization form of context-level RAG (Equation 3), we observe that most existing top-k retrieval objectives are inherited from document-level RAG and are not fully aligned with the context-level setting. The most common strategy retrieves documents with the largest relevance scores, $\boldsymbol{d}_i \in \text{top-}k(\Pr(\cdot \mid \boldsymbol{q}))$, which is adopted by many sparse and dense retrievers (Karpukhin et al., 2020; Izacard et al., 2022). Another line of work incorporates generation signals and retrieves according to $\boldsymbol{d}_i \in \text{top-}k(\Pr(\boldsymbol{a} \mid \cdot, \boldsymbol{q}))$ as in LM-supervised methods (Shi et al., 2024; Lin et al., 2023). Neither objective explicitly accounts for the context induced by the retrieved documents. In contrast, we derive a stronger retrieval objective that explicitly incorporates context and show that it is optimal for answer ranking under the context-level RAG marginalization framework. Specifically, we define the joint top-k objective as

$$\mathcal{T}_k^{\text{joint}} = \text{top-}k(\Pr(\cdot \mid \boldsymbol{q}, \mathcal{T}^*) \Pr(\boldsymbol{a} \mid \cdot, \boldsymbol{q}, \mathcal{T}^*)) \tag{4}$$

The intuition to prove the optimality of joint top-k is straightforward: each document contributes to answer generation through the product of $\Pr(\cdot \mid \boldsymbol{q}, \mathcal{T}^*)$ and $\Pr(\boldsymbol{a} \mid \cdot, \boldsymbol{q}, \mathcal{T}^*)$. Selecting the top-$k$ largest joint contributions maximizes the retained probability mass for the target answer. Furthermore, This accumulated contribution is monotonically non-decreasing. As long as it reaches another inferior true value of the answer, this probability approximation can still guarantee that the order between answers remains unchanged. Formally, we first present a basic lemma on sums of non-negative values and then use it to establish optimality.

Lemma 1. *Let $\mathcal{P} = \{p_1, p_2, p_3, \dots\}$ be a set of non-negative real numbers sorted such that $p_1 \geq p_2 \geq p_3 \geq \cdots$. Then $\sum_{i=1}^{k} p_i \geq \sum_{p' \in \mathcal{P}'} p'$ for any subset $\mathcal{P}' \subseteq \mathcal{P}$ with $|\mathcal{P}'| = k$.*

Proof. See Appendix A. □

Definition 2 (Order Preserving)**.** *A top-k approximation $\tilde{p}(\boldsymbol{a} \mid \boldsymbol{q}, \mathcal{T}^*; \mathcal{T}_k)$ is order preserving if*

$$\Pr(\boldsymbol{a}_i \mid \boldsymbol{q}, \mathcal{T}^*) > \Pr(\boldsymbol{a}_j \mid \boldsymbol{q}, \mathcal{T}^*) \Longrightarrow \tilde{p}(\boldsymbol{a}_i \mid \boldsymbol{q}, \mathcal{T}^*; \mathcal{T}_k) > \tilde{p}(\boldsymbol{a}_j \mid \boldsymbol{q}, \mathcal{T}^*; \mathcal{T}_k) \quad \forall i \neq j$$

Theorem 1. *For a given question $\boldsymbol{q}$ and two arbitrary answers $\boldsymbol{a}_1, \boldsymbol{a}_2$ satisfying $\Pr(\boldsymbol{a}_1 \mid \boldsymbol{q}, \mathcal{T}^*) > \Pr(\boldsymbol{a}_2 \mid \boldsymbol{q}, \mathcal{T}^*)$, the joint top-$k$ objective $\mathcal{T}_k^{\text{joint}}$ requires the smallest threshold k in the worst case to achieve order preservation. Formally,*

$$k^*_{\text{joint}} \leq k^* \tag{5}$$

for any alternative top-k strategy, where

$$\tilde{p}\left(\boldsymbol{a}_1 \mid \boldsymbol{q}, \mathcal{T}^*; \mathcal{T}^{\text{joint}}_{k^*_{\text{joint}}}\right) > \Pr(\boldsymbol{a}_2 \mid \boldsymbol{q}, \mathcal{T}^*) \geq \tilde{p}\left(\boldsymbol{a}_1 \mid \boldsymbol{q}, \mathcal{T}^*; \mathcal{T}^{\text{joint}}_{k^*_{\text{joint}}-1}\right) \tag{6}$$

$$\tilde{p}(\boldsymbol{a}_1 \mid \boldsymbol{q}, \mathcal{T}^*; \mathcal{T}_{k^*}) > \Pr(\boldsymbol{a}_2 \mid \boldsymbol{q}, \mathcal{T}^*) \geq \tilde{p}(\boldsymbol{a}_1 \mid \boldsymbol{q}, \mathcal{T}^*; \mathcal{T}_{k^*-1}) \tag{7}$$

Proof. See Appendix B. □

4 Practical Joint Top-k Retrieval

Although the joint top-k objective in Equation 4 is theoretically optimal for preserving answer ordering, it is impractical to compute directly during retrieval. Its scoring function depends on the question $\boldsymbol{q}$, a candidate answer $\boldsymbol{a}$, and the comprehensive context $\mathcal{T}^*$, where both $\boldsymbol{a}$ and $\mathcal{T}^*$ are unknown at retrieval time. Direct computation introduces two major obstacles. First, retrieval would need to be repeated for many candidate answers for a single query, resulting in substantial computational overhead. In open-domain question answering, the answer space may even be unbounded. Second, if $\mathcal{T}^*$ were known or could be accurately predicted from $\boldsymbol{q}$, retrieval itself would become unnecessary, since one could directly set $\mathcal{T}_k = \mathcal{T}^*$. To address these challenges, we propose a practical approximation based on listwise learning-to-rank. Instead of explicitly computing the latent joint score during inference, the retriever is trained to rank documents according to their utility for producing the best answer under the comprehensive context.

4.1 Focusing on the Best Answer

The original joint top-k objective is designed to preserve the ordering among arbitrary candidate answers. In practice, however, RAG systems are primarily concerned with whether the final generated answer is correct. Therefore, we specialize the objective to focus on the best answer $\boldsymbol{a}^*$ and define

$$\mathcal{T}_k^{\text{joint}*} = \text{top-}k\left(\Pr(\cdot \mid \boldsymbol{q}, \mathcal{T}^*) \Pr(\boldsymbol{a}^* \mid \cdot, \boldsymbol{q}, \mathcal{T}^*)\right) \tag{8}$$

By Theorem 1, this objective is sufficient to preserve the ordering between the best answer and any competing answer. At the same time, it removes the need to enumerate or score alternative candidate answers, making the objective substantially more practical for retriever learning.

4.2 Labeling Document Ranks for Training

Since $\boldsymbol{a}^*$ and $\mathcal{T}^*$ are unavailable at retrieval time, we cast retrieval as a learning-to-rank problem. Given a question $\boldsymbol{q}$, the retriever is trained to output an ordering over candidate documents $\{\boldsymbol{d}_i\}$ that approximates the latent joint objective. For each question $\boldsymbol{q}$, we define the oracle ranking indicator of document $\boldsymbol{d}_i$ as

$$\text{RI}(\boldsymbol{d}_i, \boldsymbol{q}) = \Pr(\boldsymbol{d}_i \mid \boldsymbol{q}, \mathcal{T}^*) \Pr(\boldsymbol{a}^* \mid \boldsymbol{d}_i, \boldsymbol{q}, \mathcal{T}^*) \tag{9}$$

This indicator corresponds directly to the practical objective in Equation 8 and serves as supervision for retriever training. The first term measures document relevance under the comprehensive context and can be approximated using a pretrained embedding model (e.g., Qwen3-Embedding-8B (Yang et al., 2025)) under an energy-based formulation (LeCun et al., 2006):

$$\Pr(\boldsymbol{d}_i \mid \boldsymbol{q}, \mathcal{T}^*) \approx \frac{\exp\left(\text{sim}\left(\boldsymbol{d}_i, (\boldsymbol{q}, \mathcal{T}^*)\right)\right)}{\sum_{\boldsymbol{d}_i \in \mathcal{D}} \exp\left(\text{sim}\left(\boldsymbol{d}_i, (\boldsymbol{q}, \mathcal{T}^*)\right)\right)} \tag{10}$$

where $\text{sim}(\boldsymbol{d}, (\boldsymbol{q}, \mathcal{T}^*)) = \text{E}_\text{d}(\boldsymbol{d}) \cdot \text{E}_\text{q}(\boldsymbol{q}, \mathcal{T}^*)$, $\text{E}_\text{q}(\cdot)$ is the query encoder, and $\text{E}_\text{d}(\cdot)$ is the document encoder. The second term measures how much the document supports generation of the best answer, and is estimated using a large language model such as LLaMA (Grattafiori et al., 2024) or Qwen (Yang et al., 2025):

$$\Pr(\boldsymbol{a}^* \mid \boldsymbol{d}, \boldsymbol{q}, \mathcal{T}^*) \approx \prod_i p_\theta\left(a_i^* \mid \boldsymbol{d}, \boldsymbol{q}, \mathcal{T}^*, \boldsymbol{a}^*_{1:i-1}\right) \tag{11}$$

where $p_\theta\left(a_i^* \mid \boldsymbol{d}, \boldsymbol{q}, \mathcal{T}^*, \boldsymbol{a}_{1:i-1}^*\right)$ is the estimated probability of next token a_i^* when current input is the combination of document $\boldsymbol{d}$, question $\boldsymbol{q}$, and the part of best answer $\boldsymbol{a}_{1:i-1}^*$.

According to the Definition 1, the ground $\mathcal{T}^*$ of question $\boldsymbol{q}$ can be obtained by $\mathcal{T}^* = \arg\max_{\mathcal{T}\subseteq\mathcal{D}} \Pr(\boldsymbol{a}^* \mid \boldsymbol{q}, \mathcal{T})$, which involves $\sum_{i=1}^{|\mathcal{D}|}\binom{|\mathcal{D}|}{i} = 2^{|\mathcal{D}|} - 1$ possible combinations for each question. This combinatorial search becomes computationally prohibitive when $|\mathcal{D}|$ is large. To reduce the complexity, we construct a tractable surrogate of $\mathcal{T}^*$ by approximating the local evidence landscape associated with $\boldsymbol{q}$. In particular, we consider that a comprehensive context should not only contain sufficient evidence for deriving the optimal answer, but should also allow the generator to distinguish decisive evidence from incomplete or irrelevant information. We approximate this evidence landscape using three types of documents: **Gold documents**: documents containing decisive supporting evidence required to derive the optimal answer $\boldsymbol{a}^*$; **Related documents**: documents that are relevant to $\boldsymbol{q}$ but contain only partial or incomplete supporting evidence, such as highly ranked non-gold documents retrieved by BM25; **Negative documents**: documents that contain no evidence for answering $\boldsymbol{q}$, sampled from documents associated with other questions or randomly from the corpus. These three document types respectively capture sufficient evidence, partially informative evidence, and non-evidence. Their combination provides a tractable approximation to the evidence conditions encountered in retrieval-augmented generation: the gold documents ensure answer sufficiency, the related documents represent plausible but incomplete evidence, and the negative documents introduce irrelevant information that an ideal generator should identify and ignore. In this sense, the heuristic aims to approximate a comprehensive context by covering the spectrum of evidential relevance rather than by exhaustively enumerating document subsets. We set $|\mathcal{T}^*| = k$ and sample documents approximately evenly from the above categories.

Similar to contrastive learning in dense retrieval (Karpukhin et al., 2020; Izacard et al., 2022), we do not perform labeling over the full document corpus. Instead, for each question, we construct a small candidate subset of n documents, denoted by $\boldsymbol{D}$, where $n \ll N_{\text{doc}}$. This reduces the labeling cost from $O(N_{\text{ques}} \times N_{\text{doc}})$ to $O(N_{\text{ques}} \times n)$, making large-scale training practical. For efficiency and consistency, we reuse the documents sampled for constructing the comprehensive context $\mathcal{T}^*$. That is, for each question $\boldsymbol{q}$, the candidate set $\boldsymbol{D}$ is taken as the documents contained in $\mathcal{T}^*$, and these documents are ranked according to the oracle indicator in Equation 9. The resulting question–ranking pair serves as one supervised training instance for the downstream learning-to-rank model.

4.3 Listwise Learning to Rank

After constructing the training instances, we train the retriever using a learning-to-rank objective. A widely used loss in dense retrieval is the InfoNCE loss (van den Oord et al., 2019; Izacard et al., 2022; Karpukhin et al., 2020). Although effective, InfoNCE can be interpreted from a ranking perspective as a pointwise objective, since it primarily models the probability that an individual document is positive for a given query. It does not explicitly optimize the relative ordering among multiple candidate documents, especially among non-positive items. Our goal, however, is to learn accurate joint rankings over document lists rather than merely identify a single positive document. Therefore, we adopt a listwise learning-to-rank formulation. Specifically, we use the listwise likelihood loss (Xia et al., 2008; 2009), which enjoys strong theoretical guarantees and has been shown to outperform pointwise methods empirically (Liu, 2009):

$$L(\boldsymbol{q}, \boldsymbol{D}) = -\log \prod_{i=1}^{m} \frac{\exp\left(f(\boldsymbol{q}, \boldsymbol{d}_i)\right)}{\sum_{j=i}^{n} \exp\left(f(\boldsymbol{q}, \boldsymbol{d}_j)\right)} \tag{12}$$

Here, $\boldsymbol{D} = [\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_n]$ is a ranked list of candidate documents sorted in decreasing order of the joint ranking indicator on $\mathcal{T}^*$, where $\boldsymbol{d}_1$ is the highest-ranked document. The scoring function is defined as $f(\boldsymbol{q}, \boldsymbol{d}) = \mathrm{E}'_{\mathrm{q}}(\boldsymbol{q}) \cdot \mathrm{E}'_{\mathrm{d}}(\boldsymbol{d})$, where $\mathrm{E}'_{\mathrm{q}}(\cdot)$ and $\mathrm{E}'_{\mathrm{d}}(\cdot)$ are trainable query and document encoders initialized from Contriever (Izacard et al., 2022).[1] Given n ranked documents, the hyperparameter m determines how many top-ranked positions are explicitly emphasized during training. This reflects the practical assumption that errors near the top of the ranking are more costly than errors at lower positions. Prior theoretical analysis (Xia et al., 2009) suggests setting m to match the target retrieval cutoff k, i.e., $m = k$, in top-k retrieval tasks. Notably, when $m = 1$ and the same

[1] These encoders are not necessarily different to those used in labeling (Equation 10), Since the encoders' input are typically variable-length text

Table 1: Quality of generation.

		TREC		WebQ		NQ		TriviaQA	
		EM	R-L	EM	R-L	EM	R-L	EM	R-L
Qwen3-4B	JBR-C	**0.310**	**0.458**	**0.158**	**0.310**	**0.241**	**0.367**	**0.540**	**0.629**
	DPR	0.276	0.422*	0.140	0.305	0.217***	0.332***	0.476***	0.557***
	REPLUG	0.233*	0.388*	**0.158**	0.296	0.235	0.351**	0.008***	0.025***
	Contriever	0.241*	0.378**	0.115*	0.254**	0.206***	0.308***	0.372***	0.445***
	BM25	0.155***	0.304***	0.133	0.269*	0.201***	0.308***	0.515***	0.601***
Mistral-7B	JBR-C	0.259	0.456	**0.191**	0.376	**0.233**	**0.392**	**0.580**	**0.707**
	DPR	0.259	0.454	**0.191**	**0.381**	0.214***	0.367***	0.546***	0.667***
	REPLUG	**0.276**	**0.481**	0.176	0.344*	0.018***	0.048***	0.304***	0.395***
	Contriever	0.224	0.404*	0.173	0.337*	0.189***	0.328***	0.468***	0.591***
	BM25	0.241	0.414	0.151*	0.338*	0.196***	0.334***	0.561***	0.686***
Llama3.1-8B	JBR-C	0.267	0.436	**0.169**	**0.316**	**0.288**	**0.399**	**0.601**	**0.675**
	DPR	0.250	0.440	0.155	0.300	0.263***	0.365***	0.536***	0.604***
	REPLUG	**0.302**	**0.467**	0.155	0.271**	0.000***	0.002***	0.002***	0.018***
	Contriever	0.241	0.418	0.140	0.260**	0.244***	0.335***	0.432***	0.492***
	BM25	0.181*	0.340**	0.115**	0.244***	0.250***	0.346***	0.571***	0.645***

All methods retrieve the top-5 documents. Bold and underlining indicate the best and second-best results, respectively. R-L denotes ROUGE-L. "*" indicates significant degradation compared with JBR-C ($^*p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$). McNemar's test is used for EM and Wilcoxon signed-rank test for ROUGE-L (Smeeton et al., 2025).

candidate set $\boldsymbol{D}$ is used, the listwise likelihood loss in Equation 12 reduces exactly to the InfoNCE objective. Thus, our method can be viewed as a principled generalization of standard contrastive training from pointwise supervision to listwise ranking supervision.

5 Experiment

5.1 Experimental Setting

We evaluate our method on commonly used question answering (QA) datasets (Gao et al., 2024; Karpukhin et al., 2020; Lewis et al., 2020) of varying scales including TREC (Baudiš & Šedivý, 2015), WebQ (Berant et al., 2013), NQ (Kwiatkowski et al., 2019) and TriviaQA (Joshi et al., 2017). For each dataset, a subset of examples is sampled for hyperparameter tuning according to the dataset size. We focus on retrieval objectives rather than indexing efficiency and therefore use exact nearest-neighbor search. We use three instruction-tuned LLMs as generators: **Llama3.1-8B** (Grattafiori et al., 2024), **Qwen3-4B** (Yang et al., 2025), and **Mistral-7B** (Jiang et al., 2023). Generation quality is evaluated using **EM** and **ROUGE-L** (Lin & Och, 2004). Retrieval quality is evaluated using **Recall@**k and **NDCG@**k (Järvelin & Kekäläinen, 2002). The full detail of experiment setting can be seen in Appendix C. The complete experimental code is publicly available [2].

5.2 Quality of Generation

Table 1 reports downstream generation quality measured by EM and ROUGE-L. Overall, JBR-C achieves the strongest and most consistent performance across datasets and generators, obtaining the best results in most settings under both Llama3.1-8B and Qwen3-4B. The only notable exception is TREC under Llama3.1-8B, where REPLUG outperforms other methods. REPLUG exhibits severe instability on several large-scale settings, particularly NQ and TriviaQA under Llama3.1-8B, where performance collapses to near zero. This suggests that directly approximating generation distributions can become unreliable when the generator and retrieval space are highly complex. Contriever consistently underperforms supervised retrievers, while BM25 remains competitive on datasets with stronger lexical overlap, especially TriviaQA. Despite differences in model architecture and scale, both generators produce highly consistent relative rankings among retrievers, and EM and ROUGE-L lead to similar conclusions. These results demonstrate that JBR-C provides robust improvements across generators, datasets, and evaluation metrics.

5.3 Quality of Retrieval

Table 2 reports retrieval quality measured by Recall@5 and NDCG@5 under the Qwen3-4B setting. Overall, JBR-C achieves the best performance across nearly all datasets and metrics, consis-

[2] https://anonymous.4open.science/r/jbr-c-7F01

Table 2: Quality of Retrieval.

	TREC		WebQ		NQ		TriviaQA	
	Recall	NDCG	Recall	NDCG	Recall	NDCG	Recall	NDCG
JBR-C	**0.265**	**0.301**	**0.280**	**0.283**	**0.279**	**0.260**	**0.315**	**0.359**
DPR	0.249*	0.286**	0.268	0.272	0.250***	0.230***	0.248***	0.290***
REPLUG	0.244**	0.274*	0.204***	0.206***	0.241***	0.228***	0.000***	0.000***
Contriever	0.216**	0.260**	0.197***	0.202***	0.196***	0.186***	0.167***	0.197***
BM25	0.208**	0.227***	0.200***	0.205***	0.199***	0.202***	0.310*	0.353*

All methods retrieve the top-5 documents. Bold and underlining indicate the best and second-best results, respectively. "*" indicates significant degradation compared with JBR-C ($*p < 0.05$, $**p < 0.01$, $***p < 0.001$). Wilcoxon signed-rank tests are used for statistical significance (Smeeton et al., 2025).

Table 3: Ablation study.

	TREC		WebQ		NQ		TriviaQA		Average	
	Recall	EM	Recall	EM	Recall	EM	Recall	EM	Recall	EM
Base	0.249	0.276	0.268	0.140	0.250	0.217	0.248	0.476		
+ Joint	+3.7%	+9.4%	-15.2%	+5.1%	-75.6%	-56.0%	-59.7%	-30.9%	-36.7%	-18.1%
+ LL2R	-5.4%	-3.1%	-2.7%	+7.7%	+18.5%	+8.6%	+20.0%	+9.1%	+7.6%	+5.5%
+ Both	+6.7%	+12.5%	+4.8%	+12.8%	+11.4%	+11.1%	+27.2%	+13.4%	+12.5%	+12.5%

"+ Joint" combines joint ranking with InfoNCE objective to isolate the effect of joint supervision without listwise optimization.

tent with its strong downstream generation results. These findings suggest that the proposed joint ranking objective improves both evidence coverage and ranking quality. DPR consistently provides the strongest baseline performance, while BM25 remains competitive on TriviaQA, likely due to stronger lexical overlap between questions and relevant documents. REPLUG exhibits severe instability on TriviaQA, mirroring the collapse observed in downstream generation. Recall@5 and NDCG@5 produce highly consistent method rankings across datasets. However, retrieval quality and generation quality are not always perfectly aligned. For example, REPLUG underperforms DPR on NQ retrieval metrics while achieving stronger EM under Qwen3-4B, suggesting that standard retrieval metrics may not fully capture generation utility.

5.4 Ablation Study

Table 3 presents the ablation study of JBR-C (i.e., **+ Both**) under Qwen3-4B. Starting from a standard dense retriever (Karpukhin et al., 2020) setting trained with InfoNCE (i.e., **Base**), we incrementally introduce joint ranking supervision (i.e., **+ Joint**) and listwise learning-to-rank (i.e., **+ LL2R**). Using joint ranking labels alone substantially degrades performance on several datasets, especially NQ and TriviaQA. This is likely because InfoNCE assumes a single positive target, whereas joint ranking naturally assigns high utility to multiple documents, creating a mismatch between supervision and optimization. Replacing InfoNCE with LL2R consistently improves performance on most datasets, indicating that listwise optimization better captures relative document utility than pointwise training. Combining both components yields the strongest overall performance, improving average Recall and EM by 12.5% over the baseline. These results suggest that the gains of JBR-C arise from the synergy between joint answer-aware supervision and listwise ranking optimization.

5.5 Comparison of Different Ranking Indicator

To further investigate the contribution of each component of ranking indicator, we conduct an study on different ranking indicators, including variants with or without the best answer and comprehensive context. All other experimental settings remain unchanged except for the ranking indicator. The results on Qwen3-4B are shown in Table 4. As can be observed, the proposed joint ranking objective consistently outperforms other alternatives, demonstrating its effectiveness in practice and validating the theoretical optimal.

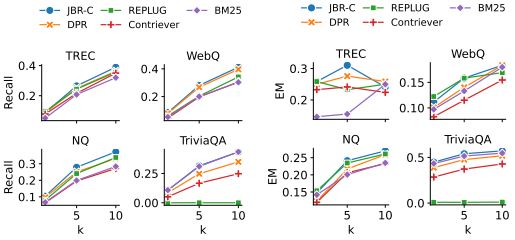
5.6 Results Under Different Top-k

We evaluate retrieval performance under different retrieval budgets with $k \in \{1, 5, 10\}$ using Qwen3-4B after training retriever on the top-5 documents. As shown in Figure 2, JBR-C consistently outperforms competing dense retrievers across most datasets and values of k. Recall@k

Table 4: Performance comparison among different ranking indicator.

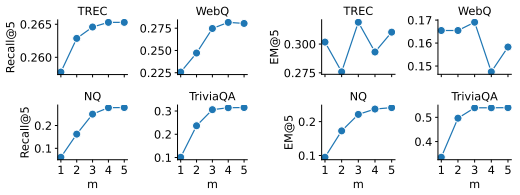
	TREC		WebQ		NQ		TriviaQA	
	EM	R-L	EM	R-L	EM	R-L	EM	R-L
JBR-C	**0.310**	**0.458**	**0.158**	**0.310**	**0.241**	**0.367**	0.540	0.629
$\Pr(\boldsymbol{d} \mid \boldsymbol{q})$	0.267	0.400	0.151	0.300	0.236	0.362	0.519	0.604
$\Pr(\boldsymbol{d} \mid \boldsymbol{q}, \mathcal{T}^*)$	0.302	0.453	0.151	0.304	0.233	0.360	**0.543**	**0.630**
$\Pr(\boldsymbol{a}^* \mid \boldsymbol{q}, \boldsymbol{d})$	0.267	0.406	0.155	0.303	0.176	0.270	0.301	0.353
$\Pr(\boldsymbol{a}^* \mid \boldsymbol{q}, \boldsymbol{d}, \mathcal{T}^*)$	0.224	0.370	0.129	0.258	0.153	0.228	0.068	0.095

Bold and underlining indicate the best and second-best results, respectively. R-L means ROUGE-L.



(a) Recall@k (b) EM@k

Figure 2: Performance under different k.



(a) Recall@5 (b) EM@5

Figure 3: JBR-C performance under different values of the listwise hyperparameter m.

increases monotonically with k, since retrieving more documents improves the chance of covering relevant evidence. In contrast, EM@k does not always improve with larger k, particularly on TREC. This suggests that better retrieval coverage does not necessarily translate into better downstream generation. From the perspective of the proposed marginalization framework, larger k should provide a better approximation of the full retrieval space. However, increasing k also introduces more irrelevant or noisy documents, which may dilute useful evidence and distract the generator. These results highlight the trade-off between evidence completeness and context quality in context-level RAG.

5.7 Hyperparameter Effect

Our method introduces the listwise likelihood hyperparameter m (Equation 12), which controls how many top-ranked documents are emphasized during training. Figure 3 reports Recall@5 and EM@5 under different values of m. Increasing m consistently improves Recall@5 across all datasets, particularly when increasing m from 1 to 2. This suggests that modeling the relative ordering among multiple relevant documents is more effective than relying on a single positive target. Compared with Recall@5, EM@5 exhibits larger fluctuations, especially on smaller datasets such as TREC and WebQ. This indicates that retrieval quality and downstream generation utility are not always perfectly aligned, and that emphasizing only the highest-ranked documents may sometimes better support generation. Overall, setting $m = k$ provides a stable trade-off between retrieval and generation performance, while maintaining consistency between training and inference.

6 Conclusion and Limitations

RAG mitigates LLM hallucinations by grounding generation in retrieved evidence. Classical formulations model answer probability through document-level marginalization over top-k candidates, whereas practical systems typically concatenate retrieved documents and generate a single answer, i.e., context-level RAG. To bridge this theory–practice gap, we introduce a marginalization framework for context-level RAG and derive a joint-based optimal retrieval objective. Since direct optimization is intractable, we propose JBR-C, a listwise learning-to-rank retriever that efficiently approximates joint ranking signals. Experiments on four QA benchmarks show that JBR-C consistently improves both retrieval and generation performance.

Limitations. Constructing joint ranking labels relies on pretrained embedding models and LLMs, incurring additional preprocessing cost and introducing dependence on external models. In addition, our evaluation focuses on dense retrieval. Extending the framework to other RAG paradigms, including hybrid retrieval, reranking, GraphRAG (Edge et al., 2025; Guo et al., 2025; Zhuang et al., 2025; Song et al., 2026), parametric RAG (Su et al., 2025), and in-context RAG (Ram et al., 2023), is an important direction for future work.

Reproducibility Statement

We make the complete experimental code publicly available at an anonymous repository[3]. The main paper and Appendix C provide the methodological details, experimental settings, datasets, evaluation protocols, and implementation details necessary to reproduce our experiments. Together with the released code, these materials are intended to facilitate independent reproduction and verification of our results.

Statement of Generative AI Use

Generative AI tools were used solely for language editing and polishing, including improving grammar, clarity, and presentation while preserving the intended technical meaning and LaTeX formatting. Example prompts include "Help me polish the following content and keep the LaTeX format." All AI-assisted text was subsequently reviewed, verified, and revised by the authors, who take full responsibility for the final content of the paper.

Ethics Statement

We acknowledge and adhere to the ICLR Code of Ethics. This work focuses on improving retrieval-augmented generation (RAG) and does not involve human-subject studies or the collection of private personal information. As with other RAG systems, potential ethical concerns may arise from the underlying language models, external knowledge sources, and their deployment, including inaccurate or biased retrieved information, privacy-sensitive content in external knowledge bases, and misuse in downstream applications. Our work does not aim to introduce or amplify such risks; rather, it studies methods for improving the reliability and effectiveness of retrieval and generation. We encourage practitioners to carefully consider the provenance, privacy, licensing, and potential biases of the data and models used when deploying such systems.

References

Shubham Agarwal, Sai Sundaresan, Subrata Mitra, Debabrata Mahapatra, Archit Gupta, Rounak Sharma, Nirmal Joshua Kapu, Tong Yu, and Shiv Saini. Cache-Craft: Managing Chunk-Caches for Efficient Retrieval-Augmented Generation. *Proc. ACM Manag. Data*, 3(3):136:1–136:28, June 2025. doi: 10.1145/3725273.

Petr Baudiš and Jan Šedivý. Modeling of the Question Answering Task in the YodaQA System. In Josanne Mothe, Jacques Savoy, Jaap Kamps, Karen Pinel-Sauvagnat, Gareth Jones, Eric San Juan, Linda Capellato, and Nicola Ferro (eds.), *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pp. 222–228, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24027-5. doi: 10.1007/978-3-319-24027-5_20.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic Parsing on Freebase from Question-Answer Pairs. In David Yarowsky, Timothy Baldwin, Anna Korhonen, Karen Livescu, and Steven Bethard (eds.), *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1533–1544, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego De Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack Rae, Erich Elsen, and Laurent Sifre. Improving Language Models by Retrieving from Trillions of Tokens. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 2206–2240, Baltimore, Maryland USA, June 2022. PMLR.

[3] https://anonymous.4open.science/r/jbr-c-7F01

Bryan Catanzaro, Zihan Liu, Wei Ping, Mohammad Shoeybi, Boxin Wang, Jiaxuan You, Yue Yu, and Chao Zhang. RankRAG: Unifying Context Ranking with Retrieval-Augmented Generation in LLMs. In *Advances in Neural Information Processing Systems 37*, pp. 121156–121184, Vancouver, BC, Canada, 2024. Neural Information Processing Systems Foundation, Inc. (NeurIPS). ISBN 979-8-3313-1438-5. doi: 10.52202/079017-3850.

Tingyang Chen, Cong Fu, Xiangyu Ke, Yunjun Gao, Yabo Ni, and Anxiang Zeng. Stitching Inner Product and Euclidean Metrics for Topology-aware Maximum Inner Product Search. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '25, pp. 2341–2350, New York, NY, USA, July 2025. Association for Computing Machinery. ISBN 979-8-4007-1592-1. doi: 10.1145/3726302.3730088.

Yung-Sung Chuang, Wei Fang, Shang-Wen Li, Wen-tau Yih, and James Glass. Expand, Rerank, and Retrieve: Query Reranking for Open-Domain Question Answering. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 12131–12147, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.768.

Zhuyun Dai, Vincent Y. Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith Hall, and Ming-Wei Chang. Promptagator: Few-shot Dense Retrieval From 8 Examples. In *The Eleventh International Conference on Learning Representations*, volume 1, pp. 1–22, Kigali Rwanda, 2023. OpenReview.net.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. THE FAISS LIBRARY. *IEEE Transactions on Big Data*, 1(1):1–17, 2025. ISSN 2332-7790. doi: 10.1109/TBDATA.2025.3618474.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From Local to Global: A Graph RAG Approach to Query-Focused Summarization, February 2025.

Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, pp. 6491–6501, New York, NY, USA, August 2024. Association for Computing Machinery. ISBN 979-8-4007-0490-1. doi: 10.1145/3637528.3671470.

Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. SPLADE v2: Sparse Lexical and Expansion Model for Information Retrieval, September 2021a.

Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. SPLADE: Sparse Lexical and Expansion Model for First Stage Ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, pp. 2288–2292, New York, NY, USA, July 2021b. Association for Computing Machinery. ISBN 978-1-4503-8037-9. doi: 10.1145/3404835.3463098.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-Augmented Generation for Large Language Models: A Survey, March 2024.

Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, Ankita Naik, Pengshan Cai, and Alfio Gliozzo. Re2G: Retrieve, Rerank, Generate. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (eds.), *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2701–2715, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.194.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, and et al. The Llama 3 Herd of Models, November 2024.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. LightRAG: Simple and Fast Retrieval-Augmented Generation, April 2025.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval Augmented Language Model Pre-Training. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 3929–3938, Vienna, Austria, November 2020. PMLR.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. *ACM Transactions on Information Systems*, 43(2):42:1–42:55, January 2025a. ISSN 1046-8188. doi: 10.1145/3703155.

Yiqian Huang, Shiqi Zhang, and Xiaokui Xiao. KET-RAG: A Cost-Efficient Multi-Granular Indexing Framework for Graph-RAG. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '25, pp. 1003–1012, New York, NY, USA, August 2025b. Association for Computing Machinery. ISBN 979-8-4007-1454-2. doi: 10.1145/3711896.3737012.

Gautier Izacard and Edouard Grave. Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 874–880, Online, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.74.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised Dense Information Retrieval with Contrastive Learning. *Transactions on Machine Learning Research*, 1(1):1–21, May 2022. ISSN 2835-8856.

Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, October 2002. ISSN 1046-8188. doi: 10.1145/582415.582418.

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

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In Regina Barzilay and Min-Yen Kan (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1147.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense Passage Retrieval for Open-Domain Question Answering. In *EMNLP 2020*, pp. 6769–6781, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.550.

Pranjal Kumar. Large language models (LLMs): Survey, technical frameworks, and future challenges. *Artificial Intelligence Review*, 57(10):260, August 2024. ISSN 1573-7462. doi: 10.1007/s10462-024-10888-y.

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. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics*, 7:453–466, August 2019. ISSN 2307-387X. doi: 10.1162/tacl_a_00276.

Carlos Lassance, Hervé Déjean, Thibault Formal, and Stéphane Clinchant. SPLADE-v3: New baselines for SPLADE, March 2024.

Yann LeCun, Sumit Chopra, Raia Hadsell, Marc'Aurelio Ranzato, and Fu Jie Huang. A Tutorial on Energy-Based Learning, 2006.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. Latent Retrieval for Weakly Supervised Open Domain Question Answering. In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 6086–6096, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1612.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474, Law St, San Diego, 2020. Curran Associates, Inc.

Xiaoxi Li, Jiajie Jin, Yujia Zhou, Yuyao Zhang, Peitian Zhang, Yutao Zhu, and Zhicheng Dou. From Matching to Generation: A Survey on Generative Information Retrieval. *ACM Trans. Inf. Syst.*, 43(3):83:1–83:62, May 2025. ISSN 1046-8188. doi: 10.1145/3722552.

Chin-Yew Lin and Eduard Hovy. Automatic Evaluation of Summaries Using N-gram Co-occurrence Statistics. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 150–157, Alberta, Canada, 2003. Association for Computational Linguistics.

Chin-Yew Lin and Franz Josef Och. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL '04, pp. 605–es, USA, July 2004. Association for Computational Linguistics. doi: 10.3115/1218955.1219032.

Xi Victoria Lin, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Richard James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvasy, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. RA-DIT: Retrieval-Augmented Dual Instruction Tuning. In *The Twelfth International Conference on Learning Representations*, volume 1, pp. 1–25, Vienna Austria, October 2023. OpenReview.net.

Tie-Yan Liu. Learning to Rank for Information Retrieval. *Foundations and Trends® in Information Retrieval*, 3(3):225–331, June 2009. ISSN 1554-0669, 1554-0677. doi: 10.1561/1500000016.

Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. Generation-Augmented Retrieval for Open-Domain Question Answering. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4089–4100, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.316.

Runwen Qiu and Jing Tang. Efficient Approximate Nearest Neighbor Search via Hemi-Sphere Centroids Graph. *Proc. ACM Manag. Data*, 3(6):321:1–321:26, December 2025. doi: 10.1145/3769786.

Ori Ram, Gal Shachaf, Omer Levy, Jonathan Berant, and Amir Globerson. Learning to Retrieve Passages without Supervision. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (eds.), *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2687–2700, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.193.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. In-Context Retrieval-Augmented Language Models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331, November 2023. ISSN 2307-387X. doi: 10.1162/tacl_a_00605.

Stephen Robertson. Understanding inverse document frequency: On theoretical arguments for IDF. *Journal of Documentation*, 60(5):503–520, October 2004. ISSN 0022-0418. doi: 10.1108/00220410410560582.

Stephen Robertson and Hugo Zaragoza. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, December 2009. ISSN 1554-0669, 1554-0677. doi: 10.1561/1500000019.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Richard James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. REPLUG: Retrieval-Augmented Black-Box Language Models. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 8371–8384, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.463.

Nigel Smeeton, Neil H. Spencer, and Peter Sprent. *Applied Nonparametric Statistical Methods*. Chapman and Hall/CRC, New York, 5 edition, March 2025. ISBN 978-0-429-32617-2. doi: 10.1201/9780429326172.

Yifan Song, Xingjian Tao, Zhicheng Yang, Yihong Luo, and Jing Tang. EHRAG: Bridging Semantic Gaps in Lightweight GraphRAG via Hybrid Hypergraph Construction and Retrieval, April 2026.

Weihang Su, Yichen Tang, Qingyao Ai, Junxi Yan, Changyue Wang, Hongning Wang, Ziyi Ye, Yujia Zhou, and Yiqun Liu. Parametric Retrieval Augmented Generation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '25, pp. 1240–1250, New York, NY, USA, July 2025. Association for Computing Machinery. ISBN 979-8-4007-1592-1. doi: 10.1145/3726302.3729957.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding, January 2019.

Liang Wang, Nan Yang, and Furu Wei. Query2doc: Query Expansion with Large Language Models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 9414–9423, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.585.

Yifan Wang, Haodi Ma, and Daisy Zhe Wang. LIDER: An efficient high-dimensional learned index for large-scale dense passage retrieval. *Proc. VLDB Endow.*, 16(2):154–166, October 2022. ISSN 2150-8097. doi: 10.14778/3565816.3565819.

Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. Listwise approach to learning to rank: Theory and algorithm. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pp. 1192–1199, New York, NY, USA, July 2008. Association for Computing Machinery. ISBN 978-1-60558-205-4. doi: 10.1145/1390156.1390306.

Fen Xia, Tie-yan Liu, and Hang Li. Statistical Consistency of Top-k Ranking. In *Advances in Neural Information Processing Systems*, volume 22, pp. 1–9, Vancouver, B.C., Canada, 2009. Curran Associates, Inc.

An Yang, Anfeng Li, Baosong Yang, and et al. Qwen3 Technical Report, May 2025.

Runjie Yu, Weizhou Huang, Shuhan Bai, Jian Zhou, and Fei Wu. AquaPipe: A Quality-Aware Pipeline for Knowledge Retrieval and Large Language Models. *Proc. ACM Manag. Data*, 3(1): 11:1–11:26, February 2025. doi: 10.1145/3709661.

Hailin Zhang, Xiaodong Ji, Yilin Chen, Fangcheng Fu, Xupeng Miao, Xiaonan Nie, Weipeng Chen, and Bin Cui. PQCache: Product Quantization-based KVCache for Long Context LLM Inference. *Proc. ACM Manag. Data*, 3(3):201:1–201:30, June 2025. doi: 10.1145/3725338.

Xi Zhao, Bolong Zheng, Xiaomeng Yi, Xiaofan Luan, Charles Xie, Xiaofang Zhou, and Christian S. Jensen. FARGO: Fast Maximum Inner Product Search via Global Multi-Probing. *Proc. VLDB Endow.*, 16(5):1100–1112, January 2023. ISSN 2150-8097. doi: 10.14778/3579075.3579084.

Xiaoyao Zhong, Haotian Li, Jiabao Jin, Mingyu Yang, Deming Chu, Xiangyu Wang, Zhitao Shen, Wei Jia, George Gu, Yi Xie, Xuemin Lin, Heng Tao Shen, Jingkuan Song, and Peng Cheng. VSAG: An Optimized Search Framework for Graph-Based Approximate Nearest Neighbor Search. *Proc. VLDB Endow.*, 18(12):5017–5030, September 2025a. ISSN 2150-8097. doi: 10.14778/3750601.3750624.

Zijie Zhong, Hanwen Liu, Xiaoya Cui, Xiaofan Zhang, and Zengchang Qin. Mix-of-Granularity: Optimize the Chunking Granularity for Retrieval-Augmented Generation. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert (eds.), *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 5756–5774, Abu Dhabi, UAE, January 2025b. Association for Computational Linguistics.

Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Haonan Chen, Zheng Liu, Zhicheng Dou, and Ji-Rong Wen. Large Language Models for Information Retrieval: A Survey. *ACM Transactions on Information Systems*, 44(1):12:1–12:54, November 2025. ISSN 1046-8188. doi: 10.1145/3748304.

Luyao Zhuang, Shengyuan Chen, Yilin Xiao, Huachi Zhou, Yujing Zhang, Hao Chen, Qinggang Zhang, and Xiao Huang. LinearRAG: Linear Graph Retrieval Augmented Generation on Large-scale Corpora, November 2025.

A Proof of Lemma 1

Proof. Sort the elements in $\mathcal{P}'$ in decreasing order as $\mathcal{P}' = \{p'_1, p'_2, \ldots, p'_k\}$. By construction, $p_i \geq p'_i$ for all $i = 1, \ldots, k$. Summing both sides yields $\sum_{i=1}^{k} p_i \geq \sum_{p' \in \mathcal{P}'} p'$. □

B Proof of Theorem 1

Proof. First, for any retrieval strategy, the top-k approximation is monotonically non-decreasing with respect to k, since each additional term is non-negative:

$$\tilde{p}\left(\boldsymbol{a} \mid \boldsymbol{q}, \mathcal{T}^*; \mathcal{T}_k\right) \geq \tilde{p}\left(\boldsymbol{a} \mid \boldsymbol{q}, \mathcal{T}^*; \mathcal{T}_{k-1}\right) \quad \forall \boldsymbol{a}, \boldsymbol{q}, k \tag{13}$$

Put the threshold k^*_{joint} into any alternative top-k strategy. Then, according to Lemma 1, there are only two cases.

Case 1. $\tilde{p}\left(\boldsymbol{a}_1 \mid \boldsymbol{q}, \mathcal{T}^*; \mathcal{T}_{k^*_{\text{joint}}}\right) > \Pr\left(\boldsymbol{a}_2 \mid \boldsymbol{q}, \mathcal{T}^*\right)$, then

$$\tilde{p}\left(\boldsymbol{a}_1 \mid \boldsymbol{q}, \mathcal{T}^*; \mathcal{T}_{k^*_{\text{joint}}}\right) > \Pr\left(\boldsymbol{a}_2 \mid \boldsymbol{q}, \mathcal{T}^*\right) \geq \tilde{p}\left(\boldsymbol{a}_1 \mid \boldsymbol{q}, \mathcal{T}^*; \mathcal{T}^{\text{joint}}_{k^*_{\text{joint}}-1}\right) \geq \tilde{p}\left(\boldsymbol{a}_1 \mid \boldsymbol{q}, \mathcal{T}^*; \mathcal{T}_{k^*_{\text{joint}}-1}\right)$$

Therefore, since k^*_{joint} complies the definition of the smallest threshold in worst case to achieve the order for the top-k approximation, it indicates that $k^* = k^*_{\text{joint}}$

Case 2. $\tilde{p}\left(\boldsymbol{a}_1 \mid \boldsymbol{q}, \mathcal{T}^*; \mathcal{T}_{k^*_{\text{joint}}}\right) \leq \Pr\left(\boldsymbol{a}_2 \mid \boldsymbol{q}, \mathcal{T}^*\right)$, then

$$\tilde{p}\left(\boldsymbol{a}_1 \mid \boldsymbol{q}, \mathcal{T}^*; \mathcal{T}_{k^*}\right) \geq \Pr\left(\boldsymbol{a}_2 \mid \boldsymbol{q}, \mathcal{T}^*\right) \geq \tilde{p}\left(\boldsymbol{a}_1 \mid \boldsymbol{q}, \mathcal{T}^*; \mathcal{T}_{k^*_{\text{joint}}}\right)$$

According to the non-decreasing property (Equation 13), we have $k^* \geq k^*_{\text{joint}}$. Combining case 1 and 2, the proof is complete. □

C Experiment Setting

Workflow overview We evaluate our method on commonly used question answering (QA) datasets of varying scales. For each dataset, a subset of examples is sampled for hyperparameter tuning according to the dataset size. Each retriever is evaluated with different generators, and performance is measured mainly from two perspectives: generation quality and relevance between question and retrieved documents. The complete experimental code is publicly available [4].

Datasets We use four widely adopted QA datasets (Gao et al., 2024; Karpukhin et al., 2020; Lewis et al., 2020), which differ in question style, answer format, and corpus scale, making them standard testbeds for RAG systems. TREC (Baudiš & Šedivý, 2015) and WebQ (Berant et al., 2013) are early QA benchmarks with relatively small corpora and short factual answers. Natural Questions (NQ) (Kwiatkowski et al., 2019) and TriviaQA (Joshi et al., 2017) are large-scale datasets designed to evaluate reading comprehension and open-domain QA. For small datasets (TREC and WebQ), we sample 200 training instances and 50 test instances for hyperparameter tuning. For larger datasets (NQ and TriviaQA), we sample 2,000 training instances and 500 test instances.

Retrievers A retriever selects a small set of candidate documents from a large corpus that may contain evidence for answering a question. Since our work focuses on retrieval objectives of embedding models rather than indexing techniques, we use brute-force exact nearest-neighbor search instead of approximate indices. We compare our method with the following representative dense and sparse retrievers: **DPR** (Karpukhin et al., 2020; Lewis et al., 2020). A popular dense retriever that retrieves the top-k documents based on relevance scores. **REPLUG** (Shi et al., 2024). A generation-aware dense retriever that aligns retrieval with generation probabilities. **Contriever** (Izacard et al., 2022). An unsupervised dense retriever trained via contrastive learning. **BM25** (Robertson & Zaragoza, 2009). A classic sparse retriever based on term-frequency and inverse-document-frequency statistics.

[4] https://anonymous.4open.science/r/jbr-c-7F01

Table 5: Summary of datasets.

Name	# Ques	# Doc	GA/Q[1]	GD/Q[2]	LS[3]
TREC (Baudiš & Šedivý, 2015)	1,241	96,224	3.4	14.8	0.08
WebQ (Berant et al., 2013)	2,752	177,857	2.6	11.1	0.07
NQ (Kwiatkowski et al., 2019)	65,395	5,752,564	1.2	8.5	0.11
TriviaQA (Joshi et al., 2017)	67,173	2,886,414	15.6	12.3	0.14

[1] Average number of ground answers per question
[2] Average number of ground documents per question
[3] Average lexical similarity between the question and ground-truth relevant documents, measured by ROUGE-1 (Lin & Hovy, 2003)

Generators A generator is a LLM that produces the final answer conditioned on the question and retrieved documents. We use two instruction-tuned LLMs with different sizes and architectures: **Llama3.1-8B** (Grattafiori et al., 2024): Llama-3.1-8B-Instruct[5]. **Qwen3-4B** (Yang et al., 2025): Qwen3-4B-Instruct-2507[6]. **Mistral-7B**: Mistral-7B-Instruct-v0.3[7].

Evaluation metric for generation To evaluate the quality of generated answers, we use the following metrics: **EM (Exact Match)**. Measures the percentage of predictions that match any ground-truth answer exactly after normalization (e.g., removing punctuation and lowercasing). This metric is typically used for short-answer or factoid questions. **ROUGE-L** (Lin & Och, 2004): Recall-Oriented Understudy for Gisting Evaluation based on Longest common subsequence. A recall-oriented metric based on the longest common subsequence between the generated text and reference answers. ROUGE-L captures fluency and informativeness, making it suitable for evaluating long-form or open-domain answers.

Evaluation metric for retrival They evaluate whether the retriever successfully identifies documents that are relevant to the questions. We report the following standard information retrieval metrics: **Recall@**k. It is calculated by dividing the number of ground-truth relevant documents found in the top-k results by the total number of ground-truth relevant documents available in the entire dataset. It measures the ability of the retriever to capture all relevant documents within a fixed candidate set. **NDCG@**k (Järvelin & Kekäläinen, 2002). Normalized Discounted Cumulative Gain at rank k, which evaluates the ranking quality by assigning higher weights to relevant documents appearing earlier in the ranked list. NDCG@k is normalized by the ideal DCG (i.e., the best possible ranking), yielding a score between 0 and 1.

Technical details For the generator component, we use the default hyperparameters of each model. The prompt templates are provided in the publicly available code we mentioned before. For the retriever component, we adopt pretrained Contriever (Izacard et al., 2022) as the backbone encoder, pretrained Qwen3-8B-Embedding, Qwen3-4B-Instruct-2507 (Yang et al., 2025), and Llama-3.1-8B-Instruct (Grattafiori et al., 2024) as the joint scoring model. The generator used for generation scoring is identical to the downstream generator and is also employed during training data preparation for joint document ranking. For each training query, we construct a ranked list of $n = 5$ documents, consisting of two ground-truth documents, two hard negatives retrieved by BM25 that are not labeled as ground truth, and one randomly sampled negative document. These documents are then jointly ranked using the proposed joint scoring mechanism to produce supervision signals for training. During training, we finetune the retriever for 50 epochs using AdamW with a learning rate of 10^{-7} and a batch size of 8, with early stopping applied. For REPLUG, its original version requires data preparation for every epoch, which is too time-consuming on large datasets. Therefore, we only prepare the data once at the beginning and it is fair to compare it with other algorithm that only require one-time preparation. Experiments were conducted on a machine equipped with one NVIDIA H100, 8 virtual CPU cores, and 2048 GB of memory.

[5] https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct
[6] https://huggingface.co/Qwen/Qwen3-4B-Instruct-2507
[7] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3

Table 6: Performance comparison among different construction of comprehensive context.

	TREC		WebQ		NQ		TriviaQA	
	EM	R-L	EM	R-L	EM	R-L	EM	R-L
JBR-C	**0.310**	**0.458**	**0.158**	**0.310**	**0.241**	**0.367**	**0.540**	**0.629**
All Gold	0.276	0.432	0.119	0.254	0.233	0.351	0.536	0.622
All Relv	0.241	0.388	0.140	0.282	0.219	0.332	0.531	0.617
All Neg	0.241	0.386	**0.158**	0.303	0.239	0.362	0.530	0.617

All methods retrieve the top-5 documents. Bold and underlining indicate the best and second-best results, respectively. R-L denotes ROUGE-L.

D Different Construction of Comprehensive Context

We evaluate the sensitivity of JBR-C to different comprehensive context construction strategies in Table 6. Specifically, we report the results with Qwen3-4B under different context construction settings as follows: **All Gold** samples all documents from the ground-truth relevant documents in the dataset, **All Relv** uses documents retrieved by BM25, and **All Neg** randomly samples documents from the document corpus. As shown above, the three alternative context construction strategies consistently underperform JBR-C. These results demonstrate that the construction of comprehensive context is an important factor affecting performance. Our current strategy, which combines samples from these three types of contexts, empirically achieves the best overall performance.

E Time Cost

Table 7: Time cost (hours)

	TREC		WebQ		NQ		TriviaQA	
	Prep	Tra	Prep	Tra	Prep	Tra	Prep	Tra
JBR-C	0.144	0.096	0.312	**0.194**	7.995	**5.203**	7.939	5.094
DPR	0.003	**0.090**	0.007	0.201	0.541	5.515	0.231	5.457
REPLUG	0.107	0.092	0.252	0.206	17.821	5.328	11.177	**4.886**
Contriever	**0.001**	0.091	**0.002**	0.199	**0.105**	5.683	**0.065**	5.181

Bold and underlining indicate the best and the second, respectively. **Prep**: time for training data preparation. **Tra**: retriever training time. BM25 is excluded as it requires no training.

Table 8: Generalization performance

	TREC		WebQ		TriviaQA	
	Recall	EM	Recall	EM	Recall	EM
JBR-C	**0.320**	0.250	**0.226**	**0.176**	0.243	**0.509**
DPR	0.228	0.241	0.186	0.162	0.152	0.412
REPLUG	0.256	**0.267**	0.194	0.151	**0.244**	0.493
Contriever	0.190	0.241	0.179	0.122	0.172	0.384

Bold and underlining indicate the best and the second, respectively. BM25 lacks a comparable "generalization" setting and is therefore excluded.

Table 7 shows the preparation cost and training time. While JBR-C has higher preparation cost than DPR and Contriever, its training time is comparable to other dense retrievers. REPLUG exhibits the highest preparation cost on large datasets due to additional indexing and retrieval steps during data preparation, whereas Contriever is the most efficient owing to its unsupervised training pipeline.

F Cross-Dataset Generalization

We evaluate cross-dataset generalization by training all retrievers on NQ and testing them on other datasets without fine-tuning. As shown in Table 8, JBR-C achieves the best overall performance and comparable to REPLUG some scenarios such as TriviaQA in Recall. These results indicate that JBR-C transfers effectively across datasets with different distributions, suggesting that the proposed joint ranking objective learns more generalizable retrieval signals rather than dataset-specific heuristics.

G Adding Reranker into the Pipeline

We additionally evaluate the impact of integrating a reranker into the RAG pipeline. Specifically, we use Qwen3-Reranker-0.6B as the reranking module and select the top-5 documents from the top-100 retrieved candidates. The Table 9 reports the results. The reranker generally improves the final generation performance, but it also reduces the performance gap introduced by different embedding models. This is because the reranker refines the ranking of retrieved candidates and partially compensates for the retrieval quality difference.

Table 9: Quality of generation after adding reranker.

	TREC		WebQ		NQ		TriviaQA	
	EM	R-L	EM	R-L	EM	R-L	EM	R-L
JBR-C (NoRerank)	**0.310**	**0.458**	0.158	0.310	0.241	0.367	0.540	0.629
JBR-C	0.259	0.406	0.169	0.346	**0.304**	**0.457**	**0.597**	**0.687**
DPR	0.267	0.415	0.173	0.346	0.297	0.449	0.581	0.670
REPLUG	0.259	0.407	0.173	0.351	0.058	0.085	0.082	0.108
Contriever	0.267	0.414	**0.180**	0.351	0.291	0.436	0.539	0.629
BM25	0.250	0.416	0.173	**0.353**	0.287	0.429	0.588	0.681

All methods retrieve the top-100 documents and select the top-5 after reranking. Bold and underlining indicate the best and second-best results, respectively. R-L denotes ROUGE-L.

It is worth noting that adding reranking makes the overall RAG pipeline approximately 2× slower during inference. This highlights the different design objectives of embedding models and reranking models. Embedding models focus on balancing retrieval accuracy and efficiency for million- or billion-scale corpora, while reranking models prioritize accuracy on a much smaller candidate set (typically dozens or hundreds of documents).

Although embedding and reranking models may share similar optimization objectives, they are fundamentally different components in the RAG pipeline. Adapting our joint ranking objective to reranking models is an interesting and promising future direction that requires further investigation.

H Case Study

Below is an example comparing JBR-C and DPR on a NQ example.

Question: Who was the creator of Victoria's Secret?

Ground-truth answer: Roy Raymond

Listing 1: Ground-truth document

```
{
"title": "Victoria's Secret",
"text": "the show.\" Victoria's Secret Victoria's Secret is an American
    designer, manufacturer, and marketer of women's lingerie, womenswear,
     and beauty products. Founded in 1977 as a response to packaged
    underwear, which the company's founder considered to be \"ugly,
    floral-print nylon nightgowns\", the company is now the largest
    American retailer of women's lingerie. Victoria's Secret was founded
    by Roy Raymond, and his wife Gaye Raymond, in San Francisco,
    California, on June 12, 1977. Eight years prior to founding Victoria'
    s Secret, in the late 1960s, Raymond was embarrassed when purchasing
    lingerie for his wife at a department store. Newsweek reported him
    looking back"
}
```

Both JBR-C and DPR retrieve the ground-truth document at rank 6. However, their top-1 retrieved documents show an important difference. DPR retrieves a document related to the Victoria's Secret brand (Gloria Vanderbilt is also a famous fashion brand) but does not contain the answer entity. In contrast, JBR-C retrieves a document that directly mentions Roy Raymond, although the document is not labeled as the ground-truth document.

Listing 2: DPR top-1 retrieved document

```
{
"title": "Gloria Vanderbilt",
"text": "Morgan Vanderbilt was an unfit parent. As an adult in the 1970s,
     Gloria Laura Vanderbilt became known in connection with a line of
```

```
    fashions, perfumes, and household goods bearing her name. She was
    particularly noted as an early developer of designer blue jeans. She
    is a member of the Vanderbilt family of New York and the mother of
    CNN television anchor Anderson Cooper. Vanderbilt was born on
    February 20, 1924, in Manhattan, New York City, the only child of
    railroad heir Reginald Claypoole Vanderbilt (1880-1925) and his
    second wife, Gloria Morgan (1904-1965). When Gloria was born, her
    father was heard"
}
```

Listing 3: JBR-C top-1 retrieved document

```
{
"title": "Victoria's Secret",
"text": "padded push-up bras\" were niche products during this period
    found \"alongside feathered boas and provocative pirate costumes at
    Frederick's of Hollywood\" outside of the mainstream product
    offerings available at department stores. In 1977, Raymond borrowed
    $40,000 from his parents and $40,000 from a bank to establish
    Victoria's Secret: a store in which men could feel comfortable buying
     lingerie. The company's first store was located in Stanford Shopping
     Center in Palo Alto, California. Raymond picked the name \"Victoria\
    " after Queen Victoria of the United Kingdom to associate with the
    refinement of the Victorian era. The \"Secret\" was what was hidden
    underneath"
}
```

This case demonstrates that JBR-C is able to retrieve documents that better support downstream answer generation, even when they are not necessarily identical to the annotated ground-truth documents. This aligns with our motivation that retrieval should optimize for answer generation rather than only document relevance.