

PURECOVER: BRIDGING THE GAP IN RE-RANKING FOR RETRIEVAL-AUGMENTED GENERATION VIA BALANCING COVERAGE AND NOISE

Anonymous authors

Paper under double-blind review

ABSTRACT

Re-ranking, originating from Information Retrieval (IR), has become a critical technique for filtering retrieved documents in Retrieval-Augmented Generation (RAG). Current RAG systems often directly apply re-rankers from traditional IR, which were originally designed to provide relevant and diverse documents to human users. However, this adoption overlooks a fundamental gap: unlike humans can use selective attention to filter noise and focus on key evidence, LLMs lack this ability. This gap causes traditional re-rankers to fail in covering essential evidence and minimizing noise for LLMs, significantly hurting RAG performance, especially in complex question-answering tasks. To address this, we argue that RAG re-rankers should serve a distinct objective: not only ensuring the coverage of key information but also minimizing noise in the selected document set. To achieve this objective, we propose **PureCover**, a document selection framework tailored for RAG. Instead of relying on traditional Top- K re-ranking, we reformulate the document selection process as a multi-objective optimization problem and solve it by exploiting LLM attention patterns during goal-oriented reasoning. To improve efficiency, we distill the selection capability into an LLM selector via a set-wise strategy. Experiments on four multi-hop QA benchmarks demonstrate that PureCover consistently outperforms state-of-the-art baselines, achieving a better balance between coverage and noise for RAG.

1 INTRODUCTION

Originating in Information Retrieval (IR), re-ranking has become a core technique for filtering retrieved documents in Retrieval-Augmented Generation (RAG) (Gao et al., 2023; Zhao et al., 2023). In traditional IR systems (e.g., search engines), re-rankers aim to present relevant and diverse documents to users, allowing them to integrate information from useful documents (Carbonell & Goldstein, 1998; Wu et al., 2024). This design primarily relies on humans’ selective attention mechanism, which enables them to actively focus on key information while ignoring query-irrelevant ones during information-seeking tasks (Müller & Krummenacher, 2006) (see Figure 1 (a) for an example).

With the rise of Large Language Models (LLMs), these re-rankers are increasingly used to select documents for LLMs in RAG systems rather than human users (Zhao et al., 2023; Gao et al., 2023). This shift exposes a critical gap: unlike human beings, who can focus on key information actively, LLMs do not exhibit such selective attention, making them struggle to focus on key information and more vulnerable to irrelevant documents (Shi et al., 2023; Cuconasu et al., 2024). Nevertheless, most RAG systems directly adopt traditional IR re-rankers (Qi et al., 2020; Glass et al., 2022; Li et al., 2024), whether relevance-oriented (Xiao et al., 2024; Zhang et al., 2025) or coverage-oriented (Santos et al., 2010; Carbonell & Goldstein, 1998), without addressing this gap.

Our empirical study finds that directly applying traditional IR re-rankers into RAG systems often leads to significant performance degradation, especially on complex question answering (QA) tasks (e.g., multi-hop QA). As shown in Figure 1 (b), relevance-oriented re-rankers (Xiao et al., 2024), which prioritize query-document relevance, often yield insufficient coverage of the essential information required to answer complex queries (Lee et al., 2025). Conversely, coverage-oriented re-rankers promote diversity across subtopics but inevitably introduce noisy documents that distract

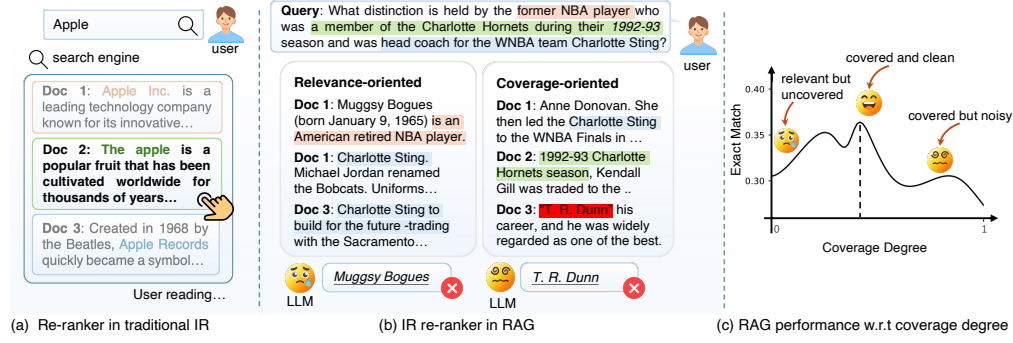


Figure 1: (a) Traditional IR re-rankers rely on humans’ selective attention: users can naturally extract key information from a diversified ranking list. For example, a user searching for “Apple” with the intent “apple nutrition facts” can naturally ignore results about “Apple Inc.” or “Apple Records” and focus on information about the fruit. (b) LLMs, however, lack this mechanism and are vulnerable to noise, making it challenging to apply traditional IR re-rankers to RAG directly. (c) This gap points to a distinct objective for RAG: unlike in IR, a re-ranker for RAG should not only ensure coverage of key information but also minimize noise in the re-ranked list.

the generator and amplify hallucinations (Chen et al., 2023) (Cossio, 2025). Experiments on HotpotQA (Yang et al., 2018) (Figure 1(c)) confirm that both strategies failed to balance the trade-off between coverage and noise, causing significant performance drops in RAG.

Based on this insight, we argue that re-ranking in RAG should adopt a distinct objective: jointly maximizing the coverage of key evidence while minimizing irrelevant noise for LLM generators. This objective fundamentally differs from the diversity-oriented coverage in traditional IR (Wu et al., 2024), motivating a new formulation. Building on this, we recast the traditional Top- K re-ranking strategy (Zehlike et al., 2017) as a multi-objective optimization problem (Deb et al., 2016) and redefine coverage and noise in the context of RAG, emphasizing the capture of essential information requirements while filtering out unhelpful documents. To address this optimization problem, we propose a novel training framework, **PureCover**, which leverages LLM attention patterns during reasoning to achieve an effective and efficient balance between coverage and noise.

Solving this multi-objective optimization problem faces two key challenges: (1) uncovering the underlying information requirements of complex queries, and (2) selecting documents that achieve broad coverage of these requirements with minimal noise. To address these challenges, we first infer the query’s information requirements through Chain-of-Thought (CoT) reasoning and estimate the associated quantities from reasoning attention patterns. By recast the objective as a binary optimization problem, we propose a greedy algorithm that efficiently maximizes coverage while minimizing noise. To improve efficiency, we further introduce a set-wise distillation method that transfers the document selection capability into an LLM selector for efficient inference. We evaluate our method on four multi-hop QA benchmarks, and the experimental results demonstrate that PureCover consistently achieves significant improvement over state-of-the-art baselines.

In summary, our contributions are as follows. (1) We emphasize a fundamental gap between the re-ranking objectives in traditional IR and RAG, which arises from the selective attention mechanism of human users. (2) To address this gap, we introduce PureCover, a novel document selection framework, which formulates document selection in RAG as a multi-objective optimization problem and leverages LLM reasoning attention patterns to effectively address this problem. (3) Extensive experiments on four multi-hop QA datasets demonstrate the effectiveness of PureCover.

2 FORMULATION

2.1 RETRIEVAL-AUGMENTED GENERATION

RAG systems aim to tackle a knowledge-intensive task (e.g., multi-hop QA), where the dataset consists of multiple query-answer pairs. Each instance consists of a raw query q paired with its corresponding ground-truth answer y . A RAG system $f(\cdot)$ takes a query q as input and returns

an answer $\hat{y}_q = f(q)$. Specifically, for each query q , the RAG process begins with a retriever $\mathcal{R}_{\text{retrieve}}$ that first retrieves a candidate set of documents $\mathcal{D}_{\text{retrieve}} = \{d_1, d_2, \dots, d_{K_1}\}$ with size K_1 . To eliminate contextually irrelevant noise and cover critical information to address the query, the initially retrieved documents $\mathcal{D}_{\text{retrieve}}$ will be further filtered by the reranker $\mathcal{R}_{\text{rerank}}$ into a smaller set $\mathcal{D}_{\text{rerank}} \subseteq \mathcal{D}_{\text{retrieve}}$. Due to the generator’s input limit, the re-ranker output is constrained by a budget $|\mathcal{D}_{\text{rerank}}| \leq K$. Finally, the filtered documents $\mathcal{D}_{\text{rerank}}$, along with the original query q , will be fed into the LLM generator as contextual information to help generate the final response $\hat{y}$.

2.2 RE-RANKER IN TRADITIONAL IR

Relevance-oriented re-ranking. The traditional relevance-oriented re-rankers, e.g., cross-encoders (Xiao et al., 2024) or LLM-based (Sun et al., 2023; Xiao et al., 2024; Zhang et al., 2025), assign an independent relevance score $\text{Rel}(q, d)$ to each document d using a relevance estimation function $\text{Rel}(\cdot)$, and select the top- K documents based on these scores:

$$\mathcal{D}_{\text{rerank}} = \{d_j | d_j \in \text{Top-}K(\text{Rel}(q, d))\}. \quad (1)$$

Coverage-oriented re-ranking. To reduce topical redundancy and address query ambiguity, traditional IR (Carbonell & Goldstein, 1998) introduces a coverage objective $\text{Cover}'(\cdot)$ alongside relevance $\text{Rel}(\cdot)$ in re-ranking. These methods aim to select the top- K documents that jointly optimize relevance and coverage in the re-ranked list, ultimately enhancing user interactions.

$$\mathcal{D}_{\text{rerank}} = \arg \max_{\mathcal{D} \subseteq \mathcal{D}_{\text{retrieve}}, |\mathcal{D}|=K} \left[\sum_{d \in \mathcal{D}} \text{Rel}(d, q) + \beta \cdot \text{Cover}'(\mathcal{D} | q) \right], \quad (2)$$

where $\text{Cover}'(\mathcal{D} | q)$ measures how well the re-ranked document set $\mathcal{D}$ covers the underlying query intents or subtopics of q (e.g., the query “Apple” may contain intents such as “Apple Inc.” or “Apple Records”), and β is a hyper-parameter that controls the trade-off between relevance and coverage.

Challenges. Both relevance-oriented and coverage-oriented re-ranking strategies rely on humans’ selective attention mechanism (Müller & Krummenacher, 2006), which enables human users to concentrate on key evidence while disregarding noise. LLMs, however, lack this mechanism. Therefore, re-ranking in RAG requires a new objective that explicitly balances covering essential information and suppressing irrelevant content for complex QA.

3 OUR APPROACH: PURECOVER

To enhance RAG performance, we optimize document selection by balancing coverage and noise, framing it as a multi-objective optimization problem (Section 3.1). We then introduce PureCover, a three-step training framework (Section 3.2): (1) prompting the LLM to reason over retrieved documents and estimate key quantities from attention signals; (2) reformulating the optimization as a binary problem and solving it efficiently with a greedy algorithm; and (3) applying set-wise distillation to transfer the selection capability into a LLM-based re-ranker. The overall workflow is illustrated in Figure 2.

3.1 OPTIMIZATION OBJECTIVE CONSTRUCTION

In this section, we construct the objective of our re-ranker to optimize the coverage and noise in the selected document set. Formally, given the initially retrieved documents $\mathcal{D}_{\text{retrieve}}$, the selection can be expressed as the following multi-objective optimization problem:

$$\mathcal{D}_{\text{rerank}} = \arg \max_{\mathcal{D} \subseteq \mathcal{D}_{\text{retrieve}}, |\mathcal{D}| \leq K} (\text{Cover}(\mathcal{D} | q), -\text{Noise}(\mathcal{D} | q)) \quad (3)$$

where $\text{Cover}(\mathcal{D} | q)$ measures how well the selected documents collectively cover all the information requirements to answer the query q , and $\text{Noise}(\mathcal{D} | q)$ measures the document noise in the selected document set. Next, we formally define coverage and noise in the context of complex QA tasks.

For a complex or compositional user query q , there are typically multiple underlying information requirements that must be satisfied to correctly answer the query (Lee et al., 2025). We denote these requirements as $\mathcal{E}_q = \{e_1, e_2, \dots, e_k\}$, obtained via an information need identification mechanism.

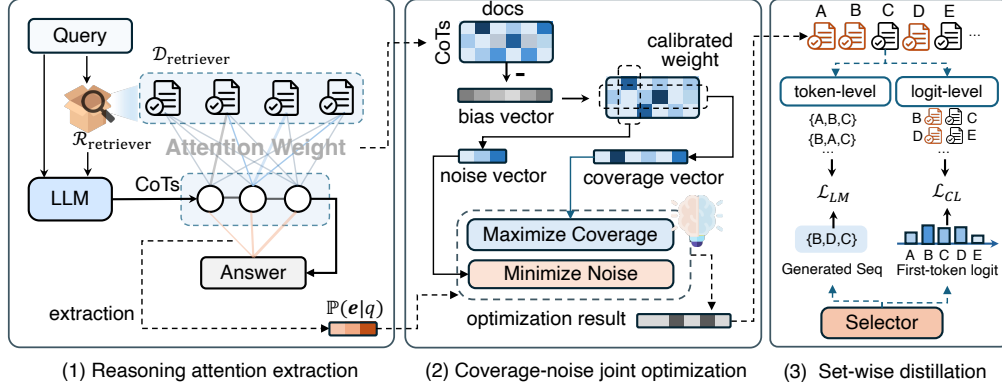


Figure 2: The overall workflow of PureCover consists of three stages: (1) We conduct a reasoning process to identify the information requirement, and extract the attention pattern; (2) We leverage the attention pattern to solve the multi-objective optimization problem; (3) To enable efficient inference, we distill the optimized result into a student model using a set-wise distillation.

Each requirement $e \in \mathcal{E}_q$ is associated with a weight $\mathbb{P}(e|q)$, such that $\sum_{e \in \mathcal{E}_q} \mathbb{P}(e|q) = 1$, representing its relative importance in addressing the query.

Information coverage. We define the information coverage of a selected subset $\mathcal{D}$ as the weighted expected probability that each information requirement e is addressed by at least one document:

$$\text{Cover}(\mathcal{D} | q) = \sum_{e \in \mathcal{E}_q} \mathbb{P}(e | q) \cdot \mathbb{P}(E_{\mathcal{D},e}), \quad \text{where } \mathbb{P}(E_{\mathcal{D},e}) = \left[1 - \prod_{d \in \mathcal{D}} (1 - \mathbb{P}(d | e)) \right], \quad (4)$$

where $E_{\mathcal{D},e}$ denotes the event that information requirement e is satisfied by at least one document in $\mathcal{D}$, $\mathbb{P}(d|e)$ is the likelihood that document d provides relevant content to satisfy the information requirement e , reflecting the document’s utility for that specific need. This definition reflects the intuitive notion that the selected document set provides high coverage if it collectively addresses all information requirements for answering the query.

Document noise. Including documents that are not helpful to answer the query can substantially distract the generator and degrade the performance of RAG systems (Shi et al., 2023; Niu et al., 2025). To capture this, we define the noise of the selected set $\mathcal{D}$ as the overall disutility (i.e., unhelpfulness) within the set:

$$\text{Noise}(\mathcal{D} | q) = \sum_{d \in \mathcal{D}} \text{DisUtil}(d | q), \quad \text{where } \text{DisUtil}(d | q) = 1 - \mathbb{P}(d | e_d^*) \mathbb{P}(e_d^* | q), \quad (5)$$

where $e_d^* = \arg \max_{e \in \mathcal{E}_q} \mathbb{P}(d|e) \mathbb{P}(e|q)$ denotes the information requirement that document d most significantly helps satisfy for answering the query. The disutility is high if this document poorly addresses the requirement or if the requirement itself is unlikely to be helpful to the query.

3.2 OBJECTIVE LEARNING AND OPTIMIZATION

To solve the optimization problem in Equation 3, we leverage the attention pattern of LLMs to effectively estimate the necessary variables required in the objective. Building on this estimation, we further propose a greedy algorithm to efficiently solve the problem, and distill the results into a lightweight selector for efficient inference.

3.2.1 LEARNING VARIABLES VIA REASONING ATTENTION

In the optimization objective (Equation 3), two key quantities are required: the information requirement $\mathbb{P}(e|q)$ and the degree to which each document supports it $\mathbb{P}(d|e)$. While prior studies have shown that LLMs’ attention patterns can highlight relevant information within long contexts (Bensemann et al., 2022; Kozlova et al., 2024; Chen et al., 2024), we extend this insight by leveraging reasoning attention patterns to explicitly estimate these two quantities. This enables us to extract the underlying information requirements of a query and evaluate document support in a way tailored

for RAG. Specifically, given a training query q and its retrieved documents $\mathcal{D}_{\text{retrieve}}$, we prompt the LLM to perform goal-oriented reasoning over these documents until producing the final answer a : $(\mathcal{C}_q, a) = \text{LLM}[P(q, \mathcal{D}_{\text{retrieve}})]$ where $\mathcal{C}_q = \{c_1, c_2, \dots, c_N\}$ is the reasoning sequence, with each step c_i corresponding to an information requirement e_i . Prompt details P in Appendix F.

Attention extraction. We leverage the token-level attention weights $\mathbf{A}$ recorded during the reasoning process as a crucial signal for downstream document selection. (See Appendix D for computational details of $\mathbf{A}$). Specifically, the token-level attention weight $\mathbf{A}_{ij}$ quantifies how much token i attends to token j . To aggregate this information at the CoT step and document level, we define the attention from each reasoning step $c \in \mathcal{C}_q$ to each retrieved document $d_k \in \mathcal{D}_{\text{retrieve}}$ as the average token-level attention between the tokens in the step and the tokens in the document: $\text{Attn}(c, d_k) = \frac{1}{|c||d_k|} \sum_{i \in c} \sum_{j \in d_k} \mathbf{A}_{ij}$, where $|c|$ and $|d_k|$ denote the number of tokens in CoT step c and document d_k , respectively.

Calibration. Raw attention weights are known to be susceptible to position biases in long contexts (Wu et al., 2025; Wan et al., 2025), such as the *lost-in-the-middle* effect (Liu et al., 2023). Using uncalibrated attention directly can therefore lead to inaccurate estimation of CoT-to-document scores. To address this, we adopt a calibration procedure following prior work (Liu et al., 2023) to remove the influence of position bias. Concretely, we define a precomputed position bias using fixed dummy CoT steps $\tilde{\mathcal{C}}$ and dummy documents $\tilde{d}_k$ and employ it to calibrate the raw attention weight assigned from CoT step c to document d_k located at position k :

$$\text{CalAttn}(c, d_k) = \text{Attn}(c, d_k) - \text{Bias}(k), \quad \text{where} \quad \text{Bias}(k) = \frac{1}{|\tilde{\mathcal{C}}|} \sum_{c \in \tilde{\mathcal{C}}} \text{Attn}(c, \tilde{d}_k), \quad (6)$$

where $\text{CalAttn}(c, d_k)$ is the position-calibrated attention from CoT step c to document d_k . Since $\tilde{\mathcal{C}}$ and $\tilde{d}_k$ are fixed, $\text{Bias}(k)$ can be precomputed once and reused for all queries, efficiently removing position-dependent distortions.

Estimation. After calibration, we leverage the attention signal to estimate the two key variables (i.e., $\mathbb{P}(e_i|q)$ and $\mathbb{P}(d|e_i)$). We define the attention-based likelihood as the normalized attention weight from answer a to each CoT step c_i (corresponding to e_i) and CoT step to each document d :

$$\begin{cases} \mathbb{P}_{\text{attn}}(e_i | q) = \frac{\exp(\text{Attn}(a, c_i)/\tau_1)}{\sum_{m=1}^{K_1} \exp(\text{Attn}(a, c_m)/\tau_1)}, & c_i \in \mathcal{C}_q, \\ \mathbb{P}_{\text{attn}}(d | e_i) = \frac{\exp(\text{CalAttn}(c, d)/\tau_2)}{\sum_{d' \in \mathcal{D}_{\text{retrieve}}} \exp(\text{CalAttn}(c, d')/\tau_2)}, & d \in \mathcal{D}_{\text{retrieve}}, c_i \in \mathcal{C}_q, \end{cases} \quad (7)$$

where τ_1 and τ_2 is the temperature coefficient that controls the sharpness of the distribution by emphasizing documents and information requirements with higher attention weights.

3.2.2 COVERAGE-NOISE JOINT OPTIMIZATION

After estimating the unknown variables in the objective function, we aim to efficiently solve the optimization problem in Equation 3. To achieve this, we formulate the objective as a 0-1 integer optimization problem in Theorem 1.

Theorem 1 (Document selection as 0-1 integer optimization). *By defining the coverage function and noise function in Equations 4 and 5, the denoised coverage-aware document task in Equation 3 can be formulated as a 0-1 integer (binary) multi-objective optimization problem:*

$$\begin{aligned} \max_{\mathbf{x}} \quad & \underbrace{\sum_{i=1}^{|\mathcal{E}|} \mathbf{e}_i \left[1 - \prod_{d=1}^{K_1} (1 - \mathbf{W}_{d,i} \mathbf{x}_d) \right]}_{\text{Coverage}} - \underbrace{\lambda \sum_{j=1}^{K_1} (1 - \max(\mathbf{W}_j \odot \mathbf{e})) \mathbf{x}_j}_{\text{Noise}} \\ \text{s.t.} \quad & \sum_i \mathbf{x}_i \leq K, \quad \mathbf{x}_i \in \{0, 1\} \quad \forall i \in [1, 2, \dots, K_1], \end{aligned} \quad (8)$$

where $\mathbf{x}$ is the binary decision vector that determines each retrieved document $d_i \in \mathcal{D}_{\text{retrieve}}$ is selected or not, i.e., $\mathbf{x}_i = 1$ if it is added to the re-ranking set $\mathcal{D}_{\text{rerank}}$, otherwise $\mathbf{x}_i = 0$. The constraint

requires that the number of selected documents does not exceed the budget K . $\mathbf{e}_i = \mathbb{P}_{attn}(e_i|q)$ and $\mathbf{W}$ denotes the document–requirement adjacency matrix, with entries $\mathbf{W}_{i,j} = \mathbb{P}_{attn}(d_i|e_j)$. $\max(\cdot)$ returns the largest entry in the input vector.

Proposition 1. *The objective function defined in Equation 8 is monotone and submodular. Therefore, the corresponding maximization problem is NP-hard.*

The proofs of Theorem 1 and Proposition 1 are provided in Appendix E. Based on Proposition 1, this optimization problem is NP-hard. Motivated by the classical result of Nemhauser et al. (1978), which guarantees that a greedy algorithm achieves a $(1 - 1/e)$ approximation for monotone submodular maximization under a cardinality constraint, we design a greedy re-ranking algorithm tailored to our coverage-noise objective. At each iteration, the algorithm selects the document that yields the largest marginal gain in the objective function (Equation 8) and adds it to the re-ranked set, as detailed in Algorithm 1.

3.2.3 SET-WISE DISTILLATION

Direct LLM reasoning is often computationally prohibitive and slow for real-world re-ranking applications. To overcome this efficiency barrier, we introduce a set-wise distillation, which transfers the complex, optimized document selection objective into a more efficient LLM-based selector for fast inference. For a query q and its retrieved documents $\mathcal{D}_{\text{retrieve}}$, we leverage teacher attention (via Equation 8 and Algorithm 1) to obtain an optimized subset $\mathcal{D}_{\text{rerank}} = \{d_1, \dots, d_m\}$. Let t_i denote the identifier of d_i and s_i the first-token logit from the student model. The selector (student model) is trained using a composite loss that captures this set-valued selection:

- *Permutation-invariant language modeling loss* $\mathcal{L}_{\text{LM}}$: Since the target is a set, the loss is minimized over all possible valid sequences $\mathcal{Y}$, which contains all permutations of the target document identifier sequence y : $\mathcal{L}_{\text{LM}} = -\min_{y' \in \mathcal{Y}} \log p_{\theta}(y' | q, \mathcal{D}_{\text{retrieve}})$
- *Contractive learning loss* $\mathcal{L}_{\text{CL}}$: This loss enforces a clear distinction between the selected documents ($d_i \in \mathcal{D}_{\text{rerank}}$) and the rejected ones ($d_j \in \mathcal{D}_{\text{retrieve}} \setminus \mathcal{D}_{\text{rerank}}$) by maximizing the score difference (logit difference $s_i - s_j$) for positive pairs $\mathcal{P}$: $\mathcal{L}_{\text{CL}} = -\frac{1}{|\mathcal{P}|} \sum_{(i,j) \in \mathcal{P}} \log \sigma(s_i - s_j)$, where $\mathcal{P}$ is the set of positive and negative document pairs: $\mathcal{P} = \{(i,j) | d_i \in \mathcal{D}_{\text{rerank}}, d_j \in \mathcal{D}_{\text{retrieve}} \setminus \mathcal{D}_{\text{rerank}}\}$. The document selector is trained with the final objective $\mathcal{L}_{\text{Final}} = \mathcal{L}_{\text{CL}} + \lambda \mathcal{L}_{\text{LM}}$.

Inference. At inference, candidate documents in $\mathcal{D}_{\text{retrieve}}$ are efficiently selected based on the first-token logits s_j , retaining only those with high logit value, i.e., $\mathcal{D}_{\text{rerank}} = \{d_j \in \mathcal{D}_{\text{retrieve}} | s_j > \tau_3\}$. This strategy leverages the learned attention patterns to prioritize high-utility documents, eliminates the need for a fixed top- K selection, and effectively filters out redundant or misleading content, thereby reducing unnecessary token consumption.

4 EXPERIMENT

4.1 EXPERIMENT SETUPS

Dataset & metrics. We evaluate our method on four widely used multi-hop QA datasets, including HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020) and MusiQue (Trivedi et al., 2022) and StrategyQA (Geva et al., 2021). Following standard RAG evaluation protocols (Yu et al., 2024), we use Exact Match (EM), F1 score (F1), and Accuracy (Acc) as our evaluation metrics.

Baselines. We compare PureCover with a broad range of baselines, including **dense retriever** e5-base-v2 (Wang et al., 2024), **relevance-oriented re-rankers**: bge-reranker-large (Xiao et al., 2024), Qwen3-Reranker-8B (Zhang et al., 2025), RankLlama (Ma et al., 2024), RankVicuna (Pradeep et al., 2023a), RankZephyr (Pradeep et al., 2023b), FIRST (Reddy et al., 2024), RankGPT (Sun et al., 2023), **coverage-oriented re-rankers**: IA-Select (Agrawal et al., 2009), MMR (Carbonell & Goldstein, 1998), xQuAD (Santos et al., 2010), and **re-rankers for RAG**: RADIO (Jia et al., 2024), SETR (Lee et al., 2025). More baseline details are provided in Appendix B.

Implementation details. We build PureCover on Qwen2.5-32B (Team, 2024) as the teacher and Qwen2.5-7B as the student model. For evaluation, we fix the retriever (e5-base-v2(Wang et al., 2024)) and generator (Qwen2.5-7B-Instruct). For a fair comparison, retrieval-only baselines retrieve the top-5 documents, while re-ranking methods operate under a re-ranking budget of 5, selecting

Retrieval Method	Model	HotpotQA		2Wiki		MusiQue		StrategyQA	
		EM	F1	EM	F1	EM	F1	EM	Acc
e5-base-v2	RETREIVAL ONLY								
	-	31.60	41.68	28.20	35.14	10.20	18.22	66.40	67.60
	CROSS-ENCODER RE-RANKER								
	bge-reranker-large	36.60	48.02	30.60	37.97	12.80	21.88	69.80	71.60
	RADIO	34.60	46.01	28.80	36.96	10.20	20.36	67.00	69.60
	LLM-BASED RE-RANKER								
	Qwen3-Reranker-8B	36.60	47.45	29.00	36.23	13.20	21.91	69.60	71.40
	RankVicuna	36.60	46.88	28.60	35.86	13.00	21.95	70.00	71.60
	RankLlama	35.20	45.57	29.80	38.87	12.40	21.14	69.20	70.20
	RankZephyr	32.80	43.18	29.20	36.84	14.20	<u>22.99</u>	68.80	71.00
	FIRST	35.80	47.40	29.00	36.54	12.80	21.86	68.60	70.40
	ICR	35.40	46.59	28.60	35.59	13.00	20.81	67.80	69.80
	RankGPT(gpt-4o)	36.80	<u>48.05</u>	<u>32.80</u>	41.03	13.80	22.86	69.40	71.20
	COVERAGE-AWARE RE-RANKER								
	IA-Select	34.00	43.66	28.20	35.60	8.80	18.13	69.80	70.80
	MMR	35.60	47.34	30.60	37.99	13.00	22.00	68.20	70.80
	xQuAD	37.00	48.01	31.80	39.55	12.60	20.52	68.20	70.20
	SETR	34.60	45.62	31.00	38.22	<u>14.60</u>	22.61	<u>70.20</u>	<u>72.00</u>
	OURS								
	PureCover	38.00	48.64	33.40	<u>40.80</u>	14.80	23.79	70.60	72.20

Table 1: End-to-end question answering results across various re-ranking models. Each model applies re-ranking or selection over the top-20 documents retrieved using *e5-base-v2*. The **bold** and underlined indicate the best and second-best performances respectively.

	HotpotQA		2WikiMultiHopQA		MusiQue		StrategyQA	
	EM	F1	EM	F1	EM	F1	EM	Acc
PureCover (ours)	38.00	48.64	33.40	40.80	14.80	23.79	70.60	72.20
ABALTION MODELS								
w/o Attention	35.40	46.60	25.40	33.62	10.40	18.90	70.00	71.00
w/o CoT weighting	35.80	47.03	31.00	38.23	13.40	22.07	69.60	70.40
w/o Position Bias Calibration	33.80	44.51	31.40	38.93	11.60	22.46	69.80	<u>71.20</u>
w/o Optimization (Avg)	<u>36.00</u>	<u>46.87</u>	31.60	39.80	<u>14.60</u>	<u>23.66</u>	68.80	69.80
w/o Optimization (Max)	34.60	45.80	<u>32.80</u>	<u>40.78</u>	14.20	23.03	69.60	<u>71.20</u>

Table 2: Ablation study of the proposed method, PureCover. The **bold** and underlined indicate the best and second-best performances, respectively.

these from the top-20 retrieved results. Additional details are in Appendix B. Our code is available at <https://anonymous.4open.science/r/PureCover-2723>.

4.2 EXPERIMENTAL RESULTS

We present our experimental results in Table 1. For coverage-oriented methods, we tune the trade-off coefficient λ and report the best results. Based on these results, we have the following observations: our method consistently outperforms both relevance-oriented and coverage-aware baselines. Relevance-oriented re-rankers (e.g., LLM-based re-rankers) emphasize document-level relevance but overlook coverage, often failing to provide sufficient evidence for complex QA tasks. Although traditional coverage-aware methods improve list-level coverage, their diversity-oriented coverage often introduces unhelpful noise and leads to suboptimal performance. While coverage-oriented LLM re-rankers rely on the model’s ability to select documents, they exhibit unstable performance on HotpotQA and 2Wiki. In contrast, our method explicitly balances coverage and noise via optimization, delivering substantial RAG improvements on complex queries.

4.3 FURTHER ANALYSIS

Ablation study. To assess the effectiveness of each component in PureCover, we conduct ablation studies on four multi-hop datasets (Table 2). We test the following variants: (1) *w/o Attention*: instead of using reasoning-attention signals to estimate $\mathbb{P}(d|e)$ (see Equation 7), we compute it based on the similarity between each CoT step and the document using BGE (Xiao et al., 2024); (2) *w/o CoT Weighting*: treat all information requirements as equally important, i.e., $\mathbb{P}(e|q) = 1$; (3) *w/o Position Bias Calibration*: use raw document-CoT attention weights without calibration; (4) *w/o Optimization*: replace the proposed greedy solver with simple heuristics—Top- K by average atten-

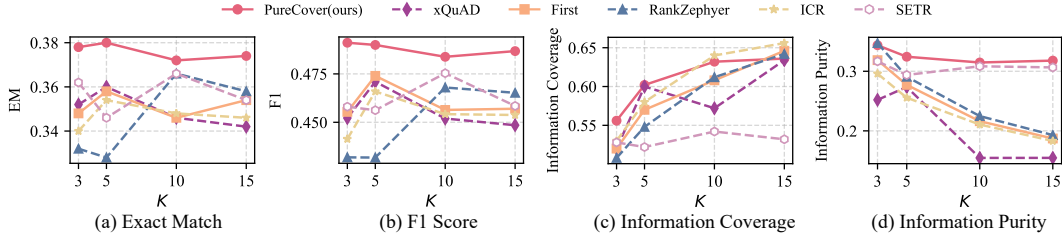


Figure 3: Performance of methods under different re-ranking budgets K on HotpotQA dataset. The Information Coverage and Information Purity metrics measure the extent of coverage and noiselessness in the re-ranked set, with detailed formulas provided in Appendix B.

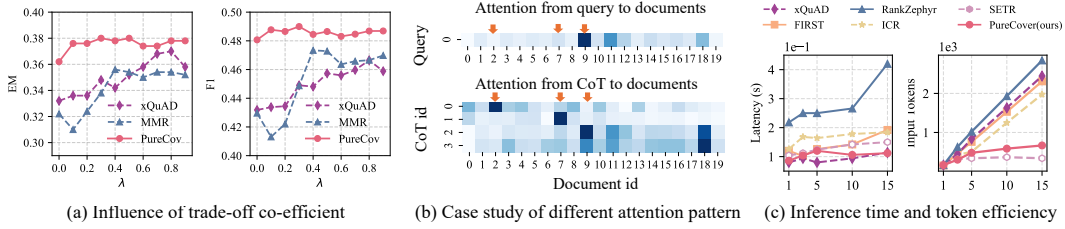


Figure 4: (a) Performance changes with different trade-off coefficients (λ); (b) Case study of attention patterns in identifying key evidence documents; (c) Inference and token efficiency.

tion (*Avg*) or by maximum single-step weight (*Max*). Results show that removing any component consistently harms performance: ignoring position bias calibration misestimates document importance, while dropping the solver yields incomplete evidence coverage and degrades RAG accuracy.

Can PureCover balance coverage and noise under varying re-ranking budget K ? Beyond EM and F1 scores, we introduce *Information Coverage* and *Information Purity* metrics to measure the presence of key evidence and the noise in the document set, respectively. As shown in Figure 3, our method PureCover achieves high coverage of key information even with small K (e.g., $K = 3$), while effectively suppressing unhelpful noise even with large K (e.g., $K = 15$). Traditional top- K re-rankers show a significant increase in noise (decrease in information purity) as the budget increases, which compromises RAG performance (EM) despite high coverage. In contrast, while SETR can select a relatively clean (unnoisy) set under large budgets, its low information coverage limits the generator’s ability to answer complex questions, resulting in suboptimal RAG performance. Our method effectively balances these two critical factors.

Influence of trade-off coefficient λ . A sensitivity analysis of the noise–coverage coefficient λ compares our method with traditional coverage-aware re-ranker (i.e., MMR and xQuAD) on HotpotQA. A smaller λ prioritizes coverage, while a larger λ emphasizes relevance (noiselessness). Figure 4 (a) illustrates that traditional baselines perform poorly at low λ (high coverage), as their diversity-based coverage introduces excessive noise that hurts accuracy. In contrast, our method is more robust and stable, effectively balancing coverage and noise and maintaining strong performance across various λ settings. We can also observe the impact of noise on our method: prioritizing either full coverage ($\lambda=0$, EM=0.362) or full noiselessness ($\lambda=1$, EM=0.372) does not yield optimal results. Instead, balancing these two objectives gives our method its best performance ($\lambda=0.3$, EM=0.38, F1=0.490), demonstrating its ability to balance noise and coverage.

Effectiveness of attention pattern during reasoning. To validate the effectiveness of our method, which uses attention from goal-oriented reasoning steps, we visualize its attention heatmap and compare it to a query-based attention baseline (i.e., ICR). As shown in Figure 4(b), the baseline method shows poor coverage of key documents, often getting distracted by noise (e.g., Documents 11 and 18). This limited focus hinders its ability to adequately cover all relevant information for complex queries. In contrast, our method aggregates attention across different CoT reasoning steps, successfully focusing on key documents. This significantly improves the coverage of the re-ranked list, ensuring all necessary information is considered to answer the query effectively.

Inference time and token efficiency. We conducted experiments to evaluate the superiority of our method’s inference efficiency and LLM generator token costs. As shown in Figure 4 (c), point-wise methods like RankZephyr exhibit a significant increase in inference time and token costs as the re-ranking length budget K grows from 1 to 15. In contrast, our method not only maintains comparable inference efficiency to traditional IR methods (i.e., xQuAD) but also achieves substantial savings in input token costs for the RAG generator component.

Other experimental analysis. Due to space constraints, more experiments of alternative retrieval methods, teacher models, re-ranking budget K , and case studies are deferred to Appendix C.

5 RELATED WORK

RAG. RAG has become a crucial technique for mitigating hallucinations and improving factual accuracy in LLMs. Most studies focused on optimizing the integration of LLMs with retrieval modules (Gao et al., 2023; Zhao et al., 2023). For example, some work has explored dynamic retrieval to determine if and when retrieval is needed (Jeong et al., 2024). Others have tackled complex questions by iteratively decomposing them (Kim et al., 2023; Sarthi et al., 2024). Another line of research focused on developing IR models better suited for RAG, such as using PPO (Schulman et al., 2017) to train a document selector (Ke et al., 2024) or building a cross-encoder re-ranker to find documents aligned with the query’s rationale (Jia et al., 2024). Despite these advances, most of these methods prioritize relevance over a crucial list-level coverage (Xie et al., 2024; Es et al., 2024), which significantly limits their effectiveness in answering complex, multi-hop queries.

Coverage-aware Re-ranking. Coverage-aware re-ranking, a well-established area in traditional IR, aims to ensure ranked results span diverse subtopics or intents (Wu et al., 2024; Clarke et al., 2008). Heuristic methods like MMR (Carbonell & Goldstein, 1998) and xQuAD (Santos et al., 2010) greedily select documents that are complementary to those already chosen. While non-heuristic learning-to-rank approaches can directly optimize coverage metrics (Yan et al., 2021), they are often limited by the high cost of human annotations. A few recent works highlight the importance of coverage in RAG (Es et al., 2024; Xie et al., 2024; Lee et al., 2025). For example, SETR (Lee et al., 2025) directly distilled the coverage ability from teacher models. However, previous studies have emphasized that increasing coverage inevitably introduces noise (Chen et al., 2023). Existing methods overlook this factor, preventing them from maintaining strong RAG performance under large re-ranking budgets.

LLM as Re-ranker. LLMs exhibit strong zero-shot ranking abilities due to their vast world knowledge, leading to the development of various ranking methods: point-wise (Zhang et al., 2025; Ma et al., 2024; Sun et al., 2023; Pradeep et al., 2023a), pair-wise (Qin et al., 2023), and list-wise (Reddy et al., 2024; Chen et al., 2024). Recent works have introduced faster decoding strategies (Reddy et al., 2024) or used calibrated attention for ranking (Chen et al., 2024). However, all of these LLM-based re-rankers follow the traditional IR Top- K re-ranking paradigm, aiming to rank the most query-relevant documents at the top. This approach overlooks crucial factors like coverage and noise, makes them difficult to apply effectively in RAG scenarios.

6 CONCLUSION

This work identifies and addresses a key limitation of existing RAG systems: the direct application of re-ranking methods designed for human users to LLMs. We highlight that this direct adoption overlooks a crucial gap: unlike humans, LLMs lack robust selective attention to filter noise and focus on key evidence. To bridge this gap, we introduce PureCover, a novel document selection framework tailored for RAG. Moving beyond the limitations of traditional Top- K re-ranking, PureCover formulates document selection as a multi-objective optimization problem, dynamically balancing information coverage and noise minimization. Our approach uniquely leverages the internal attention patterns of LLMs during goal-oriented reasoning to precisely identify and select key evidence. For practical deployment, we developed an efficient set-wise distillation strategy to transfer this sophisticated selection capability to compact LLM selectors. Extensive experiments on four multi-hop QA benchmarks reveals that PureCover consistently outperforms state-of-the-art baselines, demonstrating its ability to achieve a superior balance of coverage and noise, thereby significantly enhancing RAG performance on complex tasks.

REFERENCES

- Abdelrahman Abdallah, Bhawna Piryani, Jamshid Mozafari, Mohammed Ali, and Adam Jatowt. Rankify: A comprehensive python toolkit for retrieval, re-ranking, and retrieval-augmented generation, 2025. URL <https://arxiv.org/abs/2502.02464>.
- Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. Diversifying search results. In Proceedings of the second ACM international conference on web search and data mining, pp. 5–14, 2009.
- Joshua Bensemann, Alex Peng, Diana Benavides-Prado, Yang Chen, Neşet Tan, Paul Michael Corballis, Patricia Riddle, and Michael J Witbrock. Eye gaze and self-attention: How humans and transformers attend words in sentences. In Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics, pp. 75–87, 2022.
- Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, pp. 335–336, 1998.
- Shijie Chen, Bernal Jiménez Gutiérrez, and Yu Su. Attention in large language models yields efficient zero-shot re-rankers. arXiv preprint arXiv:2410.02642, 2024.
- Xuanang Chen, Ben He, Kai Hui, Le Sun, and Yingfei Sun. Dealing with textual noise for robust and effective bert re-ranking. Inf. Process. Manage., 60(1), January 2023. ISSN 0306-4573. doi: 10.1016/j.ipm.2022.103135. URL <https://doi.org/10.1016/j.ipm.2022.103135>.
- Charles L.A. Clarke, Maheedhar Kolla, Gordon V. Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. Novelty and diversity in information retrieval evaluation. In Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR ’08, pp. 659–666, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605581644. doi: 10.1145/1390334.1390446. URL <https://doi.org/10.1145/1390334.1390446>.
- Manuel Cossio. A comprehensive taxonomy of hallucinations in large language models, 2025. URL <https://arxiv.org/abs/2508.01781>.
- Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. The power of noise: Redefining retrieval for rag systems. In Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 719–729, 2024.
- Kalyanmoy Deb, Karthik Sindhya, and Jussi Hakanen. Multi-objective optimization. In Decision sciences, pp. 161–200. CRC Press, 2016.
- Alexander Domahidi, Eric Chu, and Stephen Boyd. Ecos: An socp solver for embedded systems. In 2013 European control conference (ECC), pp. 3071–3076. IEEE, 2013.
- Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. Ragas: Automated evaluation of retrieval augmented generation. In Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations, pp. 150–158, 2024.
- Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. arXiv preprint arXiv:2312.10997, 2023.
- Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. Did Aristotle Use a Laptop? A Question Answering Benchmark with Implicit Reasoning Strategies. Transactions of the Association for Computational Linguistics (TACL), 2021.
- Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, Ankita Rajaram Naik, Pengshan Cai, and Alfio Gliozzo. Re2g: Retrieve, rerank, generate. arXiv preprint arXiv:2207.06300, 2022.

- Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In Donia Scott, Nuria Bel, and Chengqing Zong (eds.), *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.580. URL <https://aclanthology.org/2020.coling-main.580/>.
- Rishabh Krishnan Iyer. *Submodular optimization and machine learning: Theoretical results, unifying and scalable algorithms, and applications*. PhD thesis, 2015.
- Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C Park. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. *arXiv preprint arXiv:2403.14403*, 2024.
- Pengyue Jia, Derong Xu, Xiaopeng Li, Zhaocheng Du, Xiangyang Li, Yichao Wang, Yuhao Wang, Qidong Liu, Maolin Wang, Huifeng Guo, et al. Bridging relevance and reasoning: Rationale distillation in retrieval-augmented generation. *arXiv preprint arXiv:2412.08519*, 2024.
- Jiajie Jin, Yutao Zhu, Xinyu Yang, Chenghao Zhang, and Zhicheng Dou. Flashrag: A modular toolkit for efficient retrieval-augmented generation research. *arXiv preprint arXiv:2405.13576*, 2024.
- Zixuan Ke, Weize Kong, Cheng Li, Mingyang Zhang, Qiaozhu Mei, and Michael Bendersky. Bridging the preference gap between retrievers and llms. *arXiv preprint arXiv:2401.06954*, 2024.
- Gangwoo Kim, Sungdong Kim, Byeongguk Jeon, Joonsuk Park, and Jaewoo Kang. Tree of clarifications: Answering ambiguous questions with retrieval-augmented large language models. *arXiv preprint arXiv:2310.14696*, 2023.
- Anastasia Kozlova, Albina Akhmetgareeva, Aigul Khanova, Semen Kudriavtsev, and Alena Fenogenova. Transformer attention vs human attention in anaphora resolution. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pp. 109–122, 2024.
- Dahyun Lee, Yongrae Jo, Haeju Park, and Moontae Lee. Shifting from ranking to set selection for retrieval augmented generation. *arXiv preprint arXiv:2507.06838*, 2025.
- Xiaoxi Li, Jiajie Jin, Yujia Zhou, Yongkang Wu, Zhonghua Li, Qi Ye, and Zhicheng Dou. Retrollm: Empowering large language models to retrieve fine-grained evidence within generation, 2024. URL <https://arxiv.org/abs/2412.11919>.
- Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*, 2023.
- Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. Fine-tuning llama for multi-stage text retrieval. *SIGIR '24*, pp. 2421–2425, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704314. doi: 10.1145/3626772.3657951. URL <https://doi.org/10.1145/3626772.3657951>.
- Hermann J Müller and Joseph Krummenacher. Visual search and selective attention. *Visual Cognition*, 14(4-8):389–410, 2006.
- George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14(1):265–294, 1978.
- Jingcheng Niu, Xingdi Yuan, Tong Wang, Hamidreza Saghir, and Amir H. Abdi. Llama see, llama do: A mechanistic perspective on contextual entrainment and distraction in llms, 2025. URL <https://arxiv.org/abs/2505.09338>.
- Ronak Pradeep, Sahel Sharifmoghammad, and Jimmy Lin. Rankvicuna: Zero-shot listwise document reranking with open-source large language models. *arXiv preprint arXiv:2309.15088*, 2023a.

- Ronak Pradeep, Sahel Sharifmoghaddam, and Jimmy Lin. Rankzephyr: Effective and robust zero-shot listwise reranking is a breeze!, 2023b. URL <https://arxiv.org/abs/2312.02724>.
- Peng Qi, Haejun Lee, Oghenetegiri Sido, Christopher D Manning, et al. Answering open-domain questions of varying reasoning steps from text. *arXiv preprint arXiv:2010.12527*, 2020.
- Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Le Yan, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, et al. Large language models are effective text rankers with pairwise ranking prompting. *arXiv preprint arXiv:2306.17563*, 2023.
- Revanth Gangi Reddy, JaeHyeok Doo, Yifei Xu, Md Arafat Sultan, Deevya Swain, Avirup Sil, and Heng Ji. First: Faster improved listwise reranking with single token decoding. *arXiv preprint arXiv:2406.15657*, 2024.
- Rodrygo L.T. Santos, Craig Macdonald, and Iadh Ounis. Exploiting query reformulations for web search result diversification. In *Proceedings of the 19th International Conference on World Wide Web, WWW '10*, pp. 881–890, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781605587998. doi: 10.1145/1772690.1772780. URL <https://doi.org/10.1145/1772690.1772780>.
- Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. Raptor: Recursive abstractive processing for tree-organized retrieval, 2024. URL <https://arxiv.org/abs/2401.18059>.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context, 2023. URL <https://arxiv.org/abs/2302.00093>.
- Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. Is chatgpt good at search? investigating large language models as re-ranking agents. *arXiv preprint arXiv:2304.09542*, 2023.
- Zhongxiang Sun, Qipeng Wang, Weijie Yu, Xiaoxue Zang, Kai Zheng, Jun Xu, Xiao Zhang, Song Yang, and Han Li. Rearter: Retrieval-augmented reasoning with trustworthy process rewarding. *arXiv preprint arXiv:2501.07861*, 2025.
- Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL <https://qwenlm.github.io/blog/qwen2.5/>.
- Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multi-hop questions via single-hop question composition, 2022. URL <https://arxiv.org/abs/2108.00573>.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- David Wan, Jesse Vig, Mohit Bansal, and Shafiq Joty. On positional bias of faithfulness for long-form summarization, 2025. URL <https://arxiv.org/abs/2410.23609>.
- Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training, 2024. URL <https://arxiv.org/abs/2212.03533>.
- Haolun Wu, Yansen Zhang, Chen Ma, Fuyuan Lyu, Bowei He, Bhaskar Mitra, and Xue Liu. Result diversification in search and recommendation: A survey, 2024. URL <https://arxiv.org/abs/2212.14464>.
- Xinyi Wu, Yifei Wang, Stefanie Jegelka, and Ali Jadbabaie. On the emergence of position bias in transformers. *arXiv preprint arXiv:2502.01951*, 2025.

- Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '24*, pp. 641–649, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704314. doi: 10.1145/3626772.3657878. URL <https://doi.org/10.1145/3626772.3657878>.
- Kaige Xie, Philippe Laban, Prafulla Kumar Choubey, Caiming Xiong, and Chien-Sheng Wu. Do rag systems cover what matters? evaluating and optimizing responses with sub-question coverage. *arXiv preprint arXiv:2410.15531*, 2024.
- Le Yan, Zhen Qin, Rama Kumar Pasumarthi, Xuanhui Wang, and Michael Bendersky. Diversification-aware learning to rank using distributed representation. In *Proceedings of the Web Conference 2021*, pp. 127–136, 2021.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*, 2018.
- Hao Yu, Aoran Gan, Kai Zhang, Shiwei Tong, Qi Liu, and Zhaofeng Liu. Evaluation of retrieval-augmented generation: A survey. In *CCF Conference on Big Data*, pp. 102–120. Springer, 2024.
- Meike Zehlike, Francesco Bonchi, Carlos Castillo, Sara Hajian, Mohamed Megahed, and Ricardo Baeza-Yates. Fa* ir: A fair top-k ranking algorithm. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pp. 1569–1578, 2017.
- Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.
- Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2), 2023.

APPENDIX

A GREEDY ALGORITHM

Algorithm 1: Greedy Algorithm

Input: Document-CoT overlap matrix $\mathbf{W} \in [0, 1]^{N \times K_1}$, CoT importance vector $\mathbf{e} \in [0, 1]^N$, re-ranking list length budget K_2 , coverage-relevance trade-off weight λ , truncation threshold τ .

Output: Selected document set S .

- 1: Initialize selected document set $S = \emptyset$, remaining document set $\mathcal{R} = \mathcal{D}_{\text{retrieve}}$, evidence coverage vector $\mathbf{p} = \mathbf{0}^{K_1}$.
- 2: **while** $|S| < K_2$ **and** $g^* > \tau$ **do**
- 3: $g^* = -\infty, d^* = \emptyset$
- 4: **for** $d \in \mathcal{R}$ **do**
- 5: $\mathbf{p}' = \mathbf{e} \odot [1 - (1 - \mathbf{p}) \odot (1 - \mathbf{W}_d)]$ //Compute Coverage Vector
- 6: $\Delta_g = \sum(\mathbf{p}' - \mathbf{p})$
- 7: $g = \Delta_g - \lambda \cdot [1 - \max(\mathbf{W}_d \odot \mathbf{e})]$ //Compute Noise Vector
- 8: **if** $g > g^*$ **then**
- 9: $d^* \leftarrow d, g^* \leftarrow g, \mathbf{p}^* \leftarrow \mathbf{p}'$
- 10: **end if**
- 11: **end for**
- 12: $S \leftarrow S \cup \{d^*\}, \mathcal{R} \leftarrow \mathcal{R} \setminus \{d^*\}$ //Update Selection Set
- 13: $\mathbf{p} \leftarrow \mathbf{p}^*$
- 14: **end while**
- 15: **return** S

B ADDITIONAL EXPERIMENTAL DETAILS

B.1 IMPLEMENTATION DETAILS.

We implemented all baseline methods using the FlashRAG (Jin et al., 2024) and Rankify (Abdallah et al., 2025) library. Following previous works (Sun et al., 2025), we randomly select 500 samples from the test sets of each dataset as our final test set for all baselines and our method. We set the training epochs to 10 with a learning rate of $1e-5$. For the temperature coefficients τ_1 and τ_2 , we set them both to be $4e-5$. All experiments are conducted on four NVIDIA RTX A6000 48G GPUs.

B.2 METRICS

In addition to evaluating the performance of our document selector based on the answer quality of the downstream LLM generator using metrics like EM and F1, we further define new metrics based on recall and precision to assess how well the selector balances coverage and noise under different re-ranking list budget K . Before presenting the formal definitions, we first explain how we handle the re-ranking list length budget K . After the document selector (re-ranker) outputs the selected documents $\mathcal{D}_{\text{rerank}}$, the RAG system typically constrains the set to size K to comply with the input token limitations of the LLM generator (Lee et al., 2025). Specifically, $\mathcal{D}_{\text{rerank}}^{(K)}$ refers to the subset of documents selected under budget K (i.e., the top- K documents from $\mathcal{D}_{\text{rerank}}$), which can be formally defined as:

$$\mathcal{D}_{\text{rerank}}^{(K)} = \begin{cases} \mathcal{D}_{\text{rerank}}, & \text{if } |\mathcal{D}_{\text{rerank}}| \leq K, \\ \text{Top-}K(\mathcal{D}_{\text{rerank}}), & \text{if } |\mathcal{D}_{\text{rerank}}| > K, \end{cases} \quad (9)$$

where $\text{Top-}K(\mathcal{D}_{\text{rerank}})$ denotes the top K documents in $\mathcal{D}_{\text{rerank}}$ ranked by their re-ranking scores.

Information Coverage measures how well the selected document set under a budget K covers the necessary ground-truth evidence. Formally, it is defined as the fraction of gold documents (i.e., documents that contain the golden answer) that appear in the top- K re-ranked list.

$$\text{Information Coverage@}K = \frac{|\mathcal{D}_{\text{rerank}}^{(K)} \cap \mathcal{D}_{\text{gold}}|}{|\mathcal{D}_{\text{gold}}|} \quad (10)$$

where $\mathcal{D}_{\text{gold}}$ is the set of all documents that contain the golden answer. This metric effectively measures whether the selected documents cover all the information required to answer the question. A higher value indicates that the selected documents achieve greater coverage.

Information Purity evaluates the proportion of useful documents within the selected set under a budget K . Specifically, it is defined as the fraction of gold documents among the top- K re-ranked documents.

$$\text{Information Purity@}K = \frac{|\mathcal{D}_{\text{rerank}}^{(K)} \cap \mathcal{D}_{\text{gold}}|}{|\mathcal{D}_{\text{rerank}}^{(K)}|} \quad (11)$$

where $\mathcal{D}_{\text{gold}}$ is the set of all documents that contain the golden answer. This metric effectively measures the amount of irrelevant noise present in the selected documents. A higher value indicates that the selected documents contain less noise.

Since many RAG datasets do not provide annotations for gold documents (Jin et al., 2024; Yang et al., 2018; Ho et al., 2020), we follow prior work and use $\hat{\mathcal{D}}_{\text{gold}}$ to approximate $\mathcal{D}_{\text{gold}}$. Specifically, $\hat{\mathcal{D}}_{\text{gold}} = \{d_i \in \mathcal{D} \mid d_i \text{ contains } a_{\text{gold}}\}$, i.e., the set of documents that include the gold answer. Previous studies (Jin et al., 2024) have shown that this approach provides a reasonable estimate of a re-ranker’s performance.

B.3 DATASETS

The experiments were conducted on the following four benchmark datasets:

- **HotpotQA** (Yang et al., 2018) is a large-scale multi-hop QA dataset containing 113k question-answer pairs derived from Wikipedia. Each question requires reasoning across multiple documents, with sentence-level supporting facts annotated. The dataset features diverse query types,

including comparison questions, and emphasizes compositional reasoning and explainability, making it a widely used benchmark for multi-hop retrieval and reasoning systems.

- **2WikiMultiHopQA** (Ho et al., 2020) is a multi-hop QA dataset that integrates Wikipedia text with Wikidata triples to assess step-by-step reasoning. Each question is paired with an explicit reasoning path connecting entities across documents. The dataset evaluates compositional reasoning and requires models to leverage both unstructured text and structured knowledge for accurate answers.

- **MusiQue** (Trivedi et al., 2022) is a multi-hop QA benchmark that prevents reasoning shortcuts by requiring genuine multi-step reasoning. It contains about 25k questions constructed from linked single-hop queries, each involving 2–4 reasoning steps. The dataset emphasizes strong logical dependencies between steps and includes unanswerable variants to test robustness, making it more challenging than earlier benchmarks and highlighting a larger performance gap between humans and models.

- **StrategyQA** (Geva et al., 2021) is a multi-hop QA benchmark designed to evaluate implicit reasoning. Each question is a yes/no query requiring models to infer a series of hidden reasoning steps, with supporting evidence drawn from relevant Wikipedia paragraphs. The dataset contains 2,780 examples covering a wide range of topics and reasoning strategies, emphasizing compositional reasoning and strategy inference. Compared to humans, current models still show a substantial performance gap, highlighting the challenge of implicit multi-step reasoning.

B.4 BASELINES

We consider the following re-ranking models as baselines. They include relevance-oriented cross-encoder and LLM-based re-rankers, traditional coverage-oriented models, and a LLM-based coverage-aware re-ranker.

- **bge-reranker-large** (Xiao et al., 2024) is a lightweight cross-encoder from BAAI that scores query–passage pairs using full cross-attention, providing more accurate relevance judgments than embedding-based models. Fine-tuned on large-scale data, it is widely used to rescore top- k retrieval results. As a strong open-source baseline, it represents state-of-the-art conventional re-ranking focused on individual document relevance.

- **Qwen3-Reranker** (Zhang et al., 2025) is part of the Qwen3 Embedding series, built on the Qwen3 foundation models. The re-ranker is point-wise: it scores each document independently using the yes/no logits given a query, rather than comparing pairs or lists. It is supervised fine-tuned (SFT) on high-quality labeled data to optimize relevance ranking performance. The Qwen3-Reranker models come in different sizes (0.6B, 4B, 8B parameters) to trade off efficiency and accuracy, and have shown strong results across multilingual and retrieval benchmarks. In this paper, we use Qwen3-Reranker-8B as the baseline for comparison ¹.

- **RankLlama** (Ma et al., 2024) is a pointwise re-ranker built on the LLaMA-2-7B model. Given a query and a candidate passage, it produces a relevance score to reorder retrieved documents. It achieves strong performance in both in-domain and zero-shot settings, serving as a competitive open-source baseline for document re-ranking.

- **RankVicuna** (Pradeep et al., 2023a) is a 7B open-source listwise re-ranker built on the Vicuna-7B model. It takes a query and a list of passages as input and outputs a ranked list of passage indices. Trained with GPT-generated supervision, it achieves performance comparable to GPT-3.5 on benchmarks such as TREC DL, offering a transparent alternative to proprietary re-rankers.

- **RankZephyr** (Pradeep et al., 2023b) is a zero-shot list-wise re-ranker built on the Zephyr-7B model. Fine-tuned with GPT-4-generated rankings, it produces ordered lists of passage indices given a query and candidate passages. It achieves performance comparable to GPT-4, surpassing it on some benchmarks. Its open-source and reproducible design makes it a robust baseline for evaluating listwise re-ranking methods.

- **FIRST** (Reddy et al., 2024) is a zero-shot listwise re-ranker that frames ranking as a single-token decoding task, enabling fast and efficient passage selection. Despite its simplicity, it achieves com-

¹<https://huggingface.co/Qwen/Qwen3-Reranker-8B>

petitive performance and serves as a strong open-source baseline for evaluating the raw ranking ability of instruction-tuned LLMs. In this paper, we use the Zephyr-7B-based open-source version².

- **ICR** (Chen et al., 2024) is a zero-shot re-ranker that leverages the attention patterns of large language models to estimate document relevance without generation. Given a query and candidate passages, it extracts attention-based scores to rank documents efficiently. ICR achieves competitive performance with minimal computation, providing a fast, open-source alternative for zero-shot document re-ranking.

- **RankGPT** (Sun et al., 2023) is a GPT-4-based re-ranker accessed via OpenAI’s API, operating in a zero-shot setting to rank passages given a query. It achieves state-of-the-art performance but is closed-source, non-reproducible, and costly. RankGPT4 serves as an upper-bound baseline for evaluating the effectiveness of our approach against the strongest proprietary re-ranker. We use gpt-4o³ from OpenAI to run RankGPT.

The following are the coverage-aware baselines. Since labeled or pre-collected query subtopics are often unavailable in RAG datasets, we use GPT-4o to identify the information requirements within each query and treat them as subtopics.

- **IA-Select** (Agrawal et al., 2009) is one of the most widely used coverage-aware algorithms that attempts to maximize the probability that a user finds at least one useful result within k results. Given a set of candidate documents and a set of subtopics or intents, IA-Select greedily selects documents that provide the highest expected coverage over these subtopics.

- **xQuAD** (Santos et al., 2010) is a widely used coverage-aware re-ranking algorithm that extends IA-Select by explicitly balancing relevance and coverage. It iteratively selects documents to maximize coverage over multiple query aspects or subtopics while also considering individual document relevance.

- **Maximize Marginal Relevance (MMR)** (Carbonell & Goldstein, 1998) is a widely used coverage-aware re-ranking algorithm that balances relevance and novelty in document retrieval. It iteratively selects documents to maximize relevance to the query while minimizing redundancy with already selected documents. This approach ensures that the final set of documents provides comprehensive coverage of the query’s information needs without unnecessary repetition.

- **SETR** (Lee et al., 2025) is a set-wise re-ranker for RAG that models coverage by decomposing a query into information requirements using Chain-of-Thought reasoning and selecting a set of documents that collectively satisfy them. Different from the traditional Top- K re-ranking strategy, SETR distills the ability and knowledge from a powerful GPT-4o teacher into a smaller student model. Following the authors’ implementation, we utilize GPT-4o as the teacher model and Qwen2.5-7B as the student model, which is the same as our method, PureCover.

- **RADIO** (Jia et al., 2024) is a cross-encoder-based re-ranker for bridging the gap between retrievers and generators in RAG systems. It addresses the mismatch where retrievers select documents that may not fully support the reasoning needs of LLMs. RADIO extracts reasoning rationales from LLM outputs and uses them to guide re-ranking and retriever fine-tuning, aligning retrieval with generation needs. We utilize bge-reranker-large (Xiao et al., 2024) as the re-ranking model for RADIO.

C ADDITIONAL EXPERIMENT

C.1 EXPERIMENT ON OTHER RETRIEVAL MODEL

We further evaluate our method with BM25 as the retrieval backbone to test its robustness under different retrieved document distributions. Table 3 reports the results. Across all multi-hop QA datasets, our approach consistently achieves the best performance compared to baselines, except on StrategyQA, where it performs comparably to SETR. Notably, traditional coverage-aware methods degrade significantly on more complex datasets such as MuSiQue and StrategyQA, indicating their limited ability to handle challenging reasoning scenarios. While the LLM-based selector SETR

²https://huggingface.co/rryisthebest/First_Model

³gpt-4o in this paper refers to gpt-4o-2024-11-20 from OpenAI

Retrieval Method	Model	HotpotQA		2Wiki		MusiQue		StrategyQA	
		EM	F1	EM	F1	EM	F1	EM	Acc
BM25	RETREIVAL ONLY								
	-	30.80	40.63	28.40	34.08	7.40	16.12	62.20	65.40
	CROSS-ENCODER RE-RANKER								
	Bge-Reranker-Large	36.60	47.22	32.60	39.52	10.20	18.93	65.00	66.00
	RADIO	34.60	46.01	28.80	36.96	10.20	20.39	67.00	69.60
	LLM-BASED RE-RANKER								
	Qwen3-Reranker-8B	34.40	46.19	29.60	36.18	9.80	18.75	65.40	67.20
	RankVicuna	34.00	44.69	27.40	33.82	9.40	18.27	64.60	66.60
	RankLlama	34.40	46.06	28.40	34.97	9.80	18.74	64.60	66.80
	RankZephyr	33.20	44.93	30.80	37.28	11.20	19.77	64.00	65.80
	FIRST	33.40	45.08	30.40	37.26	11.00	19.34	64.20	67.40
	ICR	35.20	46.22	31.00	37.45	8.80	17.15	62.40	65.00
	RankGPT (gpt-4o)	35.00	46.49	<u>33.20</u>	<u>39.89</u>	10.80	19.04	64.60	66.80
	COVERAGE-AWARE RE-RANKER								
	IA-Select	34.00	43.66	28.20	35.62	8.60	17.92	60.40	63.80
	MMR	34.60	45.75	31.00	37.52	11.00	19.69	62.20	64.20
	xQuAD	34.20	45.07	28.60	34.96	8.40	17.45	62.20	64.40
	SETR	<u>36.80</u>	<u>47.36</u>	31.20	38.36	<u>13.80</u>	<u>22.65</u>	70.00	71.80
	OURS								
	PureCover	37.80	47.74	34.00	40.30	15.20	23.39	<u>69.60</u>	<u>68.00</u>

Table 3: End-to-end question answering results across various ranking models. Each model applies re-ranking or selection over the top-20 documents retrieved using *BM25*. The **bold** and underlined indicate the best and second-best performances respectively.

surpasses relevance-oriented baselines on several datasets, its gains are unstable and it performs poorly on 2WikiMultiHopQA. In contrast, our method demonstrates stable and superior improvements across all datasets.

C.2 EXPERIMENT ON DIFFERENT RE-RANKING BUDGET K

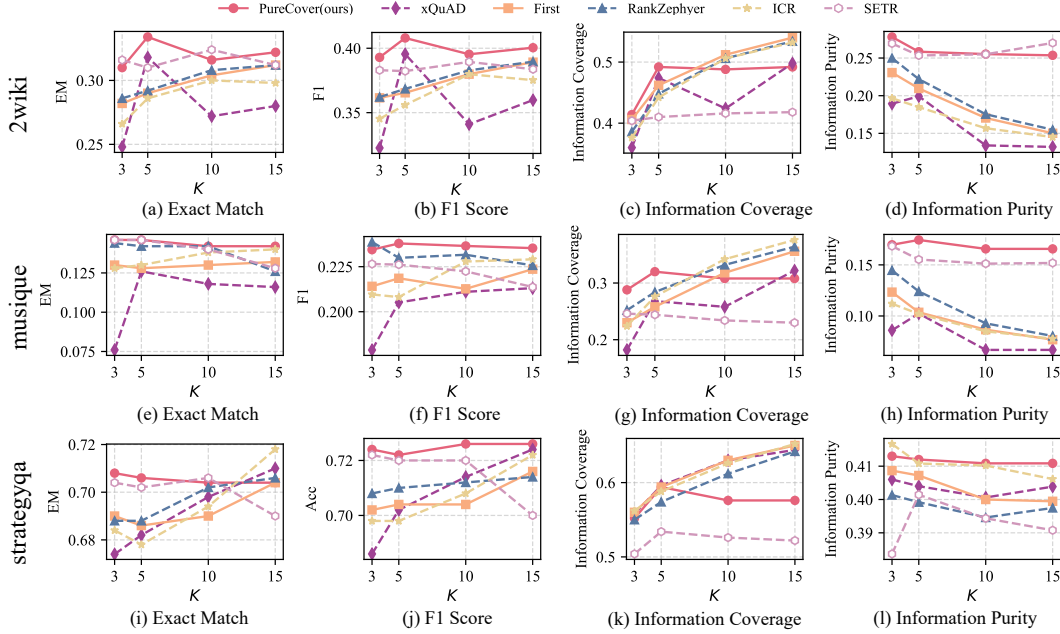


Figure 5: Performance of PureCover under varying re-ranking budgets K . Information Coverage and Information Purity are evaluated with respect to documents containing the gold answer. Subfigures (a–d), (e–h), and (i–l) correspond to experiments on 2WikiMultiHopQA, MusiQue, and StrategyQA, respectively.

As shown in Figure 5, we evaluate the performance of different methods on 2WikiMultiHopQA, MusiQue, and StrategyQA under varying re-ranking list budgets K .

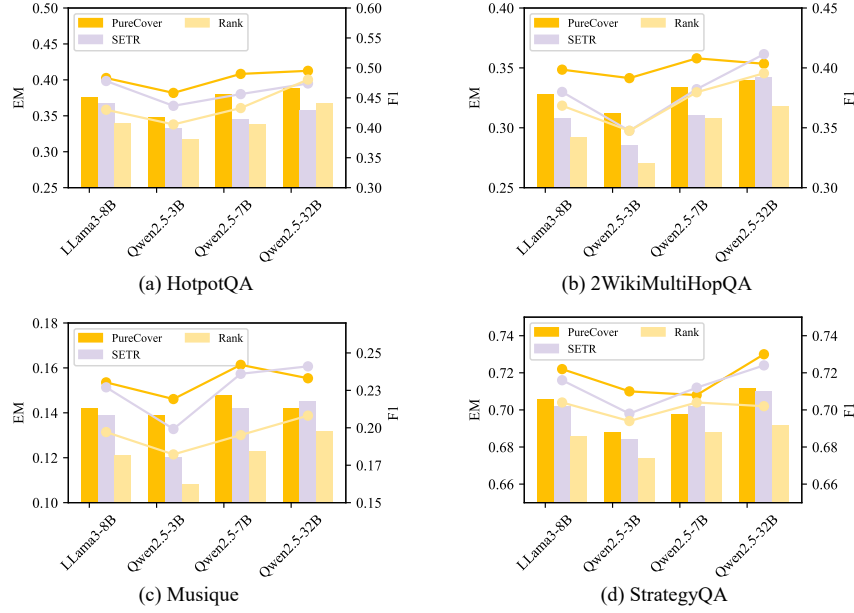


Figure 6: Ablation study on different teacher models to isolate method-level effects. The student model is applied using Qwen2.5-7B.

Overall, our method consistently achieves strong and stable results across all three datasets, maintaining high end-to-end RAG performance regardless of the budget K . In contrast, traditional Top- K re-ranking strategies (e.g., xQuAD, First, RankZephyr) tend to perform poorly when K is small (e.g., $K = 3$). This is because they struggle to ensure coverage of the critical evidence (see Figures (c), (g), and (k)). Whether based on LLMs or cross-encoders, conventional IR re-rankers suffer from insufficient coverage under low K . As K increases, these methods gradually improve coverage by leveraging relevance, but this improvement comes at the cost of introducing significant noise, which ultimately limits RAG performance. By contrast, our method is able to guarantee sufficient coverage at small K and suppress noise growth at large K , thereby sustaining relatively high performance across different budgets. It is also worth noting that SETR does not rely on a fixed Top- K list, which allows it to achieve strong noise control (e.g., in Figures (d) and (h), its information purity reaches 0.270 and 0.152 at $K = 15$). However, due to its relatively low coverage, SETR falls short of achieving optimal RAG performance.

C.3 FAIR COMPARISON ON SAME STUDENT MODEL

To ensure a fair comparison and verify that our multi-objective training approach offers methodological advantages over existing re-ranking strategies, we conduct experiments on the same student model (Qwen2.5-7B) while varying the teacher models. Specifically, we use models of different sizes and families, including LLaMA3-8B, Qwen2.5-3B, Qwen2.5-7B, and Qwen2.5-32B, as teachers. We compare our method PureCover with two strong alternatives: (1) SETR (Lee et al., 2025), which identifies query information requirements and performs document selection with the teacher model, and (2) list-wise ranking with LLMs (Rank) (Pradeep et al., 2023b). In this setting, the re-ranking budget K is fixed at 5.

Experimental results in Figure 6 show that the chosen baseline methods relying on teacher model inference for document ranking are highly dependent on model capacity and size. For example, both list-wise Rank and SETR perform poorly with smaller teachers (e.g., 3B) but significantly better with larger ones (e.g., 30B). In contrast, our approach, by combining attention with optimization rather than relying only on raw model capacity, compensates for the weaker reasoning abilities of smaller models and thus achieves competitive performance even with lightweight teachers.

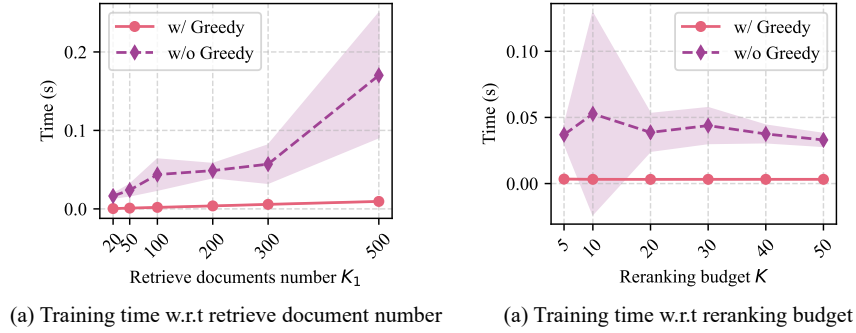


Figure 7: Training efficiency analysis under different settings: (a) number of retrieved documents K_1 , and (b) re-ranking budget K . When varying K_1 , we fix the re-ranking budget to $K = 10$; when varying K , we fix the number of retrieved documents to $K_1 = 100$. Each experiment is repeated 20 times, and we report the mean and variance.

C.4 TRAINING EFFICIENCY ANALYSIS

We propose a greedy algorithm to efficiently solve the multi-objective document selection problem, which greatly improves the training efficiency of our framework. To evaluate its effectiveness, we conduct experiments under varying numbers of retrieved documents K_1 and different re-ranking budgets K . Specifically, when analyzing the impact of K_1 , we fix the re-ranking budget to $K = 10$; when analyzing the impact of K , we set the number of retrieved documents to $K_1 = 100$. We use the **ECOS-BB** (Domahidi et al., 2013) solver to solve the original binary optimization problem in Equation 8.

Figure 7 reports the time cost of our method with and without the greedy algorithm under these two settings. For a fair comparison, we exclude the time for LLM reasoning and attention extraction, and only measure the efficiency of the optimization module. The results show that the greedy algorithm significantly accelerates training, especially when the number of retrieved documents K_1 is large. Moreover, regardless of the re-ranking budget K , our greedy-based optimizer requires nearly zero training time, while the version without greedy optimization takes 1–2 seconds per iteration. These results demonstrate that the proposed greedy algorithm not only improves scalability to large retrieval sizes but also ensures consistently efficient training across different re-ranking budgets.

C.5 CASE STUDY

To further illustrate the benefits of our proposed method, we present example cases from the HotpotQA dataset. Figure 8 shows the Top-3 retrieved documents ranked by our method, the best-performing coverage-aware baseline (xQuAD), and the LLM-based re-ranker (RankZephyr). The example query is a multi-hop question that requires multiple evidence documents as input to the generator in order to be answered correctly. For clarity, we highlight different aspects of the query and documents in different colors. And the noisy information is highlighted using the red color.

We observe that our method successfully ranks documents by considering both coverage and noise control. Specifically, the Top-3 results each capture a distinct aspect of the query, ensuring comprehensive evidence for reasoning while avoiding irrelevant content. In contrast, xQuAD achieves diversity by selecting documents about different subtopics, but these subtopics are weakly related to the actual CoT steps and often introduce noise that distracts the generator. RankZephyr, on the other hand, places the most relevant documents at the top, but due to the lack of coverage awareness, it produces a redundant list (e.g., two documents both describing NBA players), with some documents even being noisy (the document about T.R. Dunn), which ultimately fails to support the multi-hop reasoning process and distracting the LLM generator.

Example Query1: What distinction is held by the **former NBA player** who was a member of the Charlotte Hornets during their **1992-93 season** and was **head coach for the WNBA team Charlotte Sting**?

Document Reranked by PureCover: 1. Muggsy Bogues Tyrone Curtis "Muggsy" Bogues (born January 9, 1965) is an American retired basketball player. **The shortest player ever to play in the National Basketball Association.** 2. During the season, the Sting traded veteran Dawn Staley to the Houston Comets and named Charlotte basketball icon **Muggsy Bogues as their new head coach late in the season.** 3. In the 1992 NBA draft, the Hornets selected center Alonzo Mourning out of Georgetown with the second overall pick. With the addition of Mourning, **along with second-year star Larry Johnson and Muggsy Bogues** traded to the Seattle SuperSonics. Signed LaMark Baker as a free agent. Signed Lorenzo Williams as a free agent.

Response Given by PureCover: **shortest player ever to play in the National Basketball Association** ✓

Document Reranked by xQuAD: 1. **Anne Donovan. She then led the Charlotte Sting to the WNBA Finals in 2001, losing to the Los Angeles Sparks.** 2. Charlotte Sting. Michael Jordan renamed the Bobcats. Uniforms: The Charlotte Sting was one of the eight original WNBA franchises that began play in 1997. 3. **"1992-93 Charlotte Hornets season" season, Kendall Gill was traded to the Seattle SuperSonics. Signed LaMark Baker as a free agent. Signed Lorenzo Williams as a free agent.**

Response Given by xQuAD: **Anne Donovan** ✗

Document Reranked by RankZephyr: 1. Muggsy Bogues. Muggsy Bogues Tyrone Curtis "Muggsy" Bogues (born January 9, 1965) is an American retired basketball player. **The shortest player ever to play in the National Basketball Association.** 2. **T. R. Dunn** his career, and he was widely regarded as one of the best rebounding guards of the 1980s. 3. **"Charlotte Sting" to build for the future -trading with the Sacramento Monarchs for Tangelia Smith and a second-round draft pick in the 2006 draft in a deal that saw Nicole Powell traded to Sacramento.**

Response Given by RankZephyr: **Muggsy Bogues** ✗

Figure 8: Case study on HotpotQA. The query contains multiple information needs, highlighted with different background colors. The red color denotes the noise information. We compare our method against the best-performed LLM-based and coverage-aware baselines (xQuAD, RankZephyr).

D COMPUTATION OF TOKEN-LEVEL ATTENTION SCORE

In the Transformer architecture (Vaswani et al., 2017), the attention mechanism operates on three matrices: queries $\mathbf{Q} \in \mathbb{R}^{m \times d_k}$, keys, $\mathbf{K} \in \mathbb{R}^{n \times d_k}$, and values $\mathbf{V} \in \mathbb{R}^{n \times d_v}$. The attention function is defined as:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right) \mathbf{V}. \quad (12)$$

Here, each element $(\mathbf{Q}\mathbf{K}^\top)_{ij} = q_i \cdot k_j$ measures the similarity between the i -th query and the j -th key. Dividing by $\sqrt{d_k}$ controls the magnitude of the dot products, preventing overly large values that could destabilize training. The softmax function is applied row-wise, normalizing the scores into a probability distribution whose weights sum to one. These attention weights are then used to compute a weighted sum of the value vectors $\mathbf{V}$, producing the output representation.

At the token level, the attention weight

$$\mathbf{A}_{ij} = \text{softmax}\left(\frac{q_i \cdot k_j}{\sqrt{d_k}}\right)$$

represents how much token i attends to token j . These weights are computed dynamically for each input, enabling the model to capture semantic dependencies between tokens.

E PROOFS OF THEOREM 1 AND PROPOSITION 1

Proof of Theorem 1. According to previous studies of multi-objective optimization (Deb et al., 2016), the problem of balancing competing objectives can be expressed in the weighted sum form:

$$\max_{\mathcal{D}_{\text{rerank}} \subseteq \mathcal{D}_{\text{retrieve}}, |\mathcal{D}_{\text{rerank}}| \leq K} \text{Cover}(\mathcal{D}_{\text{rerank}}) - \lambda \cdot \text{Noise}(\mathcal{D}_{\text{rerank}}),$$

where λ is a trade-off coefficient between the two objectives.

Step 1: Coverage function. From Equation 4, the coverage of the selected set is:

$$\text{Cover}(\mathcal{D}_{\text{rerank}}) = \sum_{i=1}^{|\mathcal{E}|} \mathbf{e}_i \cdot \mathbb{P}(e_i \text{ covered by } \mathcal{D}_{\text{rerank}}).$$

For each information requirement e_i , the probability it is not covered by any selected document is:

$$\prod_{d \in \mathcal{D}_{\text{rerank}}} (1 - \mathbf{W}_{d,i}).$$

Thus, the probability it is covered is:

$$1 - \prod_{d \in \mathcal{D}_{\text{rerank}}} (1 - \mathbf{W}_{d,i}).$$

Introducing the binary selection variable $\mathbf{x}_d \in \{0, 1\}$, where $\mathbf{x}_d = 1$ if document d is selected, the above product becomes:

$$\prod_{d=1}^{K_1} (1 - \mathbf{W}_{d,i} \mathbf{x}_d).$$

Therefore, the coverage term is:

$$\sum_{i=1}^{|\mathcal{E}|} \mathbf{e}_i \left[1 - \prod_{d=1}^{K_1} (1 - \mathbf{W}_{d,i} \mathbf{x}_d) \right].$$

Step 2: Noise function. From Equation 5, noise is defined as:

$$\text{Noise}(\mathcal{D}_{\text{rerank}}) = \sum_{d \in \mathcal{D}_{\text{rerank}}} \text{DisUtil}(d|q).$$

Here, disutility for a document is defined as:

$$\text{DisUtil}(d|q) = 1 - \max_i \mathbf{W}_{d,i} \mathbf{e}_i.$$

Using $\mathbf{x}_d$ to indicate selection, the noise term becomes:

$$\sum_{j=1}^{K_1} (1 - \max(\mathbf{W}_j \odot \mathbf{e})) \mathbf{x}_j,$$

where $\odot$ denotes element-wise multiplication.

Step 3: Binary optimization formulation. Combining the coverage and noise terms, and introducing the budget constraint $\sum_i \mathbf{x}_i \leq K$, we obtain:

$$\begin{aligned} \max_{\mathbf{x}} \quad & \sum_{i=1}^{|\mathcal{E}|} \mathbf{e}_i \left[1 - \prod_{d=1}^{K_1} (1 - \mathbf{W}_{d,i} \mathbf{x}_d) \right] - \lambda \sum_{j=1}^{K_1} (1 - \max(\mathbf{W}_j \odot \mathbf{e})) \mathbf{x}_j, \\ \text{s.t.} \quad & \sum_i \mathbf{x}_i \leq K, \quad \mathbf{x}_i \in \{0, 1\}, \quad \forall i. \end{aligned}$$

This exactly matches the statement in Equation 8, proving that the denoised coverage-aware document selection problem can indeed be formulated as a 0–1 integer (binary) multi-objective optimization problem. $\square$

Proof of Proposition 1. We prove the claim by showing each per-evidence term is monotone and sub-modular, and then use closure properties of submodular functions. Let’s define the multi-objective objective function in Problem 8 as $g(S) = G(S) - \lambda \cdot R(S)$.

(1) Per-evidence term. Fix an evidence e_i . Define

$$h_i(S) = 1 - \prod_{d \in S} (1 - \mathbf{W}_{d,i}), \quad S \subseteq \mathcal{D}.$$

For any $S \subseteq T \subseteq \mathcal{D}$ and any $d \in \mathcal{D} \setminus T$ the marginal gain of adding d is

$$\Delta_d(S) = h_i(S \cup \{d\}) - h_i(S) = \prod_{t \in S} (1 - \mathbf{W}_{t,i}) - \prod_{t \in S} (1 - \mathbf{W}_{t,i})(1 - \mathbf{W}_{d,i}) = \mathbf{W}_{d,i} \prod_{t \in S} (1 - \mathbf{W}_{t,i}).$$

Similarly,

$$\Delta_d(T) = \mathbf{W}_{d,i} \prod_{t \in T} (1 - \mathbf{W}_{t,i}).$$

Since $0 \leq 1 - \mathbf{W}_{t,i} \leq 1$ for all t , we have

$$\prod_{t \in S} (1 - \mathbf{W}_{t,i}) \geq \prod_{t \in T} (1 - \mathbf{W}_{t,i}),$$

and therefore $\Delta_d(S) \geq \Delta_d(T)$. This is exactly the diminishing returns property, so $h_i(\cdot)$ is submodular. Moreover $\Delta_d(S) = \mathbf{W}_{d,i} \prod_{t \in S} (1 - \mathbf{W}_{t,i}) \geq 0$, so $h_i(\cdot)$ is monotone non-decreasing.

(2) Sum preserves submodularity and monotonicity for Coverage Objective. The coverage objective $G(S) = \sum_i e_i h_i(S)$ is a nonnegative linear combination of the functions h_i . Nonnegative linear combinations of submodular (resp. monotone) functions remain submodular (resp. monotone) (Iyer, 2015). Hence $G(\cdot)$ is monotone and submodular.

(3) Per-document term. Fix a document d . Define its disutility score as

$$\text{DisUtil}(d) = 1 - \max_{e \in \mathcal{E}} \mathbb{P}(d \mid e) \mathbb{P}(e \mid q).$$

Now consider the noise function over a set of documents

$$S \subseteq \mathcal{D} : R(S) = \sum_{d \in S} \text{DisUtil}(d).$$

Notice that each per-document term $c_d \cdot \text{DisUtil}(d)$ depends only on d and the evidence set $\mathcal{E}$, but not on the other documents in S . Hence, each term contributes additively and independently.

(4) Submodularity and monotonicity of Noise Objective. For any $S \subseteq T \subseteq \mathcal{D}$ and any document $d \in \mathcal{D} \setminus T$, the marginal gain of adding d is

$$\Delta_d(S) = R(S \cup \{d\}) - R(S) = \text{DisUtil}(d).$$

Similarly,

$$\Delta_d(T) = \text{DisUtil}(d).$$

Since the marginal contribution of d is identical regardless of the context set, the diminishing returns property holds trivially. This means $R(\cdot)$ is modular (a special case of submodular) (Iyer, 2015).

Moreover, as $\text{DisUtil}(d) \geq 0$, we have $\Delta_d(S) \geq 0$, implying monotonicity. Therefore, the noise objective

$$R(S) = \sum_{d \in S} \left(1 - \max_{e \in \mathcal{E}} \mathbb{P}(d \mid e) \mathbb{P}(e \mid q) \right)$$

is a monotone modular function, and hence also submodular.

(3) NP-hardness. Maximizing our proposed objective $g(\cdot)$ under a cardinality constraint is generally NP-hard. A cardinality constraint means that the solution set S is restricted to contain at most K documents (i.e., $|S| \leq K$), which reflects the practical setting of re-ranking where we can only select a limited number of documents for the generator.

This complexity includes the well-known Max-k-Cover problem as a special case. Specifically, if each document d either fully covers a particular information requirement e_i ($p_{d,i} = 1$) or does not cover it at all ($p_{d,i} = 0$), then the coverage function $h_i(S)$ becomes an indicator function that equals 1 if e_i is covered by at least one document in S , and 0 otherwise. Under this setting, maximizing $g(S)$ is equivalent to selecting K documents to cover as many weighted information requirements as possible, which is exactly the Max-k-Cover problem, which is known to be NP-hard (Iyer, 2015).

Therefore, even in the more general case where $p_{d,i}$ takes continuous values (representing partial coverage), maximizing $g(\cdot)$ remains NP-hard. This justifies the need for an efficient approximation algorithm, such as our proposed greedy approach. $\square$

You are given a query and a set of documents.
 Your goal is to reason through the documents step by step to answer the query.
Instructions:” 1. Only include reasoning steps that clearly contribute to solving the query.
 2. Make sure each step is logically connected and leads to the final answer.
 3. Use specific evidence from the documents in each step (e.g., facts, names, dates, relationships).
 4. In each step, cite document snippets or facts that directly support your reasoning.
 5. Avoid vague or redundant steps. Do not repeat the query.
 6. Limit reasoning to 5 steps or fewer. Be concise and precise.
Format: Each step must start with [Step N]: followed by your reasoning.
 Conclude with [Answer]: followed by the final answer (and nothing else).
Example format:
 [Step 1]: ...
 [Step 2]: ...
 ...
 [Answer]: ...

Table 4: Input prompt for goal-oriented reasoning of PureCover

F PROMPT DETAILS

G LLM USAGE DISCLOSURE

In accordance with ICLR 2026 policy, we disclose our use of large language models (LLMs) in preparing this manuscript. We employed GPT-5 (OpenAI) solely to aid in polishing the writing, specifically for improving clarity, grammar, and sentence structure across sections. All technical content, algorithmic contributions, experimental results, and scientific conclusions remain entirely the authors’ own work, without any LLM involvement.