Gumbel Reranking: Differentiable End-to-End Reranker Optimization

Anonymous ACL submission

Abstract

RAG systems rely on rerankers to identify relevant documents. However, fine-tuning these models remains challenging due to the scarcity of annotated query-document pairs. Existing distillation-based approaches suffer from training-inference misalignment and fail to capture interdependencies among candidate documents. To overcome these limitations, we reframe the reranking process as an attention-mask problem and propose Gumbel Reranking, an end-to-end training framework for rerankers aimed at minimizing the training-inference gap. In our approach, reranker optimization is reformulated as learning a stochastic, document-wise Top-k attention mask using the Gumbel Trick and Relaxed Top-k Sampling. This formulation enables end-to-end optimization by minimizing the overall language loss. Experiments across various settings consistently demonstrate performance gains, including a 10.4% improvement in recall on HotpotQA for distinguishing indirectly relevant documents.

1 Introduction

Retrieval-Augmented Generation (RAG) has shown great potential in natural language processing tasks (Lewis et al., 2020; Guu et al., 2020; Izacard and Grave, 2021b; Borgeaud et al., 2022). Despite their remarkable progress, retrieval models in RAG systems—comprising both the retriever and reranker—are typically trained on publicly available datasets and often struggle with long-tail queries requiring domain-specific knowledge. As a result, they necessitate further fine-tuning for specific downstream tasks (Glass et al., 2022; Shi et al., 2024). A key challenge in this context is the scarcity of labeled query-document pairs (Lee et al., 2019; Sachan et al., 2023). Therefore, a critical research question is how to end-to-end optimize the retrieval models of RAG systems solely relying on the system's final language modeling loss.

Recent efforts to improve retriever or reranker in RAG systems have explored distilling knowledge from LLMs into retrieval components. Techniques such as attention-based distillation (Izacard and Grave, 2021a) and perplexity-based distillation (Sachan et al., 2021; Shi et al., 2024; Lin et al., 2024; Izacard et al., 2023; Glass et al., 2022) have yielded notable performance gains. However, these methods still exhibit critical limitations. First, although these methods claim to be end-to-end optimized, they focus on LLM-supervised losses like KL divergence (Izacard et al., 2023; Glass et al., 2022) or marginalization (Sachan et al., 2021; Shi et al., 2024; Lin et al., 2024), which do not directly minimize the RAG system's final generation loss, leading to potential misalignment between training and evaluation objectives. Additionally, attention-based distillation suffers from the distraction problem, where accumulated attention scores do not always reflect document relevance (Ke et al., 2024; Li et al., 2024). While perplexity-based distillation methods evaluate each candidate document in isolation, neglecting the interdependencies among retrieved documents. This oversight is particularly detrimental in multi-hop reasoning tasks requiring coherent logical relationships between documents (Trivedi et al., 2022; Ho et al., 2020).

In this work, we propose a novel end-to-end strategy for training rerankers in RAG systems. We reformulate the reranking task through the lens of attention masks, where selecting the top-k subset from the retrieved candidate documents is viewed as the application of a document-wise top-k attention mask during attention computation. This perspective leads to a shift in the problem formulation: instead of directly learning a more effective reranker, we focus on learning the optimal document-wise top-k attention mask.

However, since the hard attention mask is discrete, it can not be directly optimized via gradient descent. To overcome this challenge, we introduce

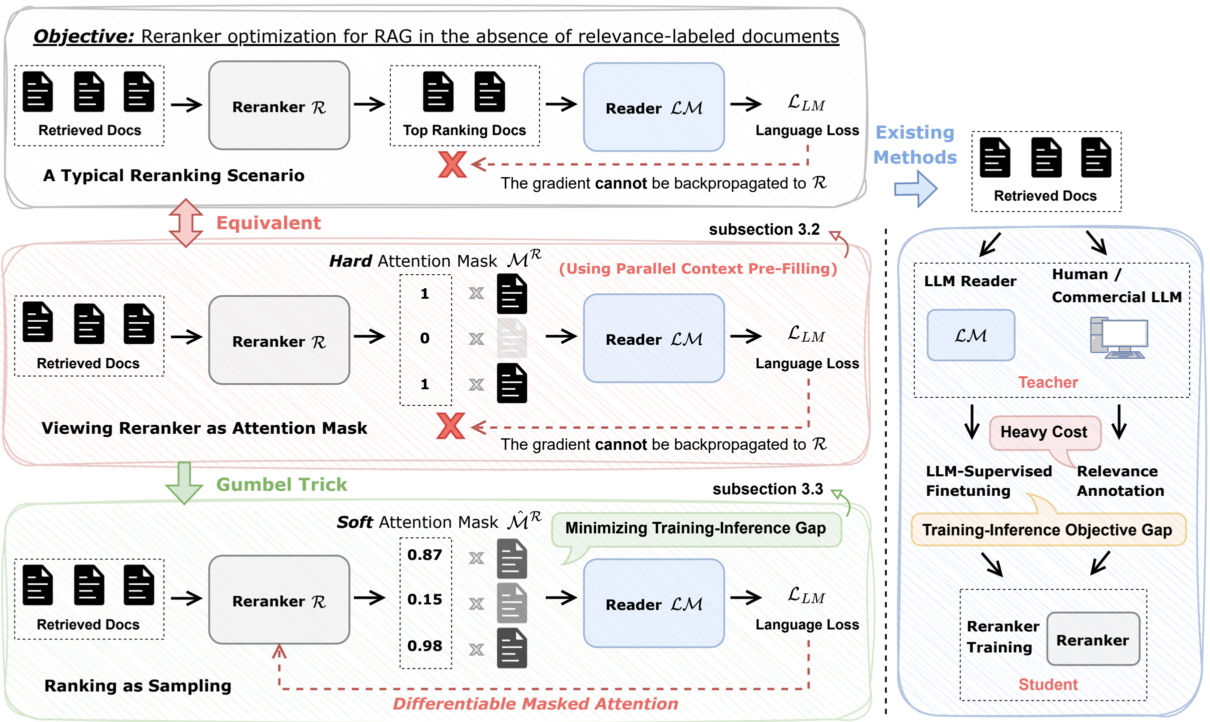

Figure 1: Vanilla reranker training methods for RAG systems typically rely on supervised learning of query-document pairs, which is limited by the scarcity of labeled data. To address this issue, existing methods leverage various LLM-supervised losses. However, this can lead to potential gaps between training and inference. In contrast, G-Rerank frames reranker training as learning a stochastic, document-wise top-k attention mask. This enables end-to-end optimization by minimizing language loss, ensuring better alignment between training and inference.

a solution based on the Gumbel Trick (Jang et al., 2017) and Relaxed Top-k techniques (Chen et al., 2018). This enables us to design a *stochastic*, top-k attention mask that is fully differentiable, allowing for end-to-end optimization. We note this approach as **D**ifferentiable **M**asked **A**ttention (DMA).

With DMA in place, we reformulate the reranking problem as learning the optimal sampling weight for the corresponding attention mask. This leads to our end-to-end training framework, which we refer to as **G**umbel **Rerank**ing (G-Rerank). Unlike previous methods that rely on LLM-supervised losses, G-Rerank directly optimizes the reranker by minimizing the overall language modeling loss of the RAG system, thereby ensuring that the training objective closely aligns with the inference process. Additionally, G-Rerank accounts for interdependencies between retrieved candidate documents, making it suitable for multi-hop QA tasks.

We evaluate our training approach across various architectures. Specifically, we conduct experiments using two language models—FiD (Izacard and Grave, 2021b) and CEPE-Llama2-7B (Yen et al., 2024)—as well as two rerankers—BGE-Reranker-Base (Xiao and Liu, 2023) and RankT5 (Zhuang et al., 2023). Our method is tested on five benchmark datasets, covering both single-hop and multi-hop QA tasks. To comprehensively assess the effectiveness of our approach, we consider three different evaluation settings: *mining*, *reranking*, and *generation*. Our proposed training strategy achieves consistent improvements across all these settings. Furthermore, compared to distillation-based methods, our training approach significantly improves the reranker's ability to distinguish *indirectly relevant documents*, leading to a 10.4% improvement in the Recall@5 metric on HotpotQA. Finally, we analyze the necessity of the Gumbel trick and the impact of prior knowledge in rerankers.

2 Related Works

Training the Reranking Module in RAG Systems The effectiveness of RAG systems relies heavily on the quality of retrieval and reranking (Glass et al., 2022; Dong et al., 2024). Traditional retrieval methods are based on lexical similarity (Robertson and Zaragoza, 2009), while recent advances leverage dense vectors and transformer architectures (Karpukhin et al., 2020; Khattab and Zaharia, 2020). However, retrieval modules fine-tuned on public datasets often require additional adaptation for specific downstream tasks (Izacard et al., 2023; Salemi and Zamani, 2024).

To bridge this gap, recent works explore fine-tuning retrieval and reranking modules for tasks such as open-domain question answering (ODQA). One common strategy distills knowledge from LLMs into retrievers by ranking candidate documents based on generated answer perplexity (Shi et al., 2024; Glass et al., 2022; Lin et al., 2024). However, such methods overlook inter-document dependencies, crucial for multi-hop reasoning tasks (Trivedi et al., 2022; Ho et al., 2020). Alternative approaches use attention scores (Izacard and Grave, 2021a) or leave-one-out methods (Izacard et al., 2023; Asai et al., 2022), but these are not end-to-end optimized for generation quality, leading to a retriever-generation gap (Ke et al., 2024).

Stochastic k-Subset Selection and Masked Attention Top-k relaxation has been widely studied for differentiable subset sampling, extending the Gumbel-Softmax trick (Jang et al., 2017; Xie and Ermon, 2019; Xie et al., 2020), with important applications in semi-structured pruning (Fang et al., 2024), model interpretability (Chen et al., 2018) and point clouds analysis (Yang et al., 2019).

The reranking process can also be modeled as a subset sampling problem. However, since retrieved documents influence LLM outputs through attention, a key challenge lies in introducing sparsity into the attention computation. Existing approaches employ soft attention masks to model discrete selections (Fan et al., 2021; Yang et al., 2019). Inspired by these methods, we model RAG reranking as a subset sampling process with soft masks, facilitating end-to-end optimization.

3 Methodology

3.1 Problem Setting

For common downstream tasks, such as Open-Domain QA (Zhu et al., 2021), the training data typically consists of an input query $\mathbf{q}$ and the corresponding ground-truth answer $\mathbf{a}$. During the retrieval process, a set of candidate documents $\mathbf{d_1}, \ldots, \mathbf{d_n}$ is retrieved based on $\mathbf{q}$. The reranker $\mathcal{R}$ is then applied to these candidate documents, generating a set of candidate scores. We retain only the top-k scored documents for further computation:

$$\begin{aligned} w_i &= \mathcal{R}(\text{Concatenate}(\mathbf{q}, \mathbf{d}_i)), \quad \forall i \in [n] \\ \mathcal{I}_k &= \{i \mid w_i \in \text{top-}k\,(\{w_i\}_{i=1}^n)\} \end{aligned} \tag{1}$$

where $[n] \triangleq \{1, 2, \ldots, n\}$. The top-$k$ documents, $\mathcal{D}_k = \{\mathbf{d}_i \mid i \in \mathcal{I}_k\}$, are selected as input to the LLM, which then computes the corresponding logits and language loss $\mathcal{L}_{LM}$. In this work, we focus on training the reranker in the RAG system. A key challenge is that the candidate documents $\mathbf{d_1}, \ldots, \mathbf{d_n}$ lack relevance annotations, making it infeasible to directly fine-tune the reranker. Additionally, although we have access to the language loss $\mathcal{L}_{LM}$, the top-k operation in Equation 1 is non-differentiable, preventing gradient propagation to the reranker and thus hindering end-to-end training.

3.2 Viewing Reranker as Attention Mask

We reinterpret the reranking process from the perspective of attention masks. Let $K_{i,t}$ and $V_{i,t}$ denote the key and value embeddings of the t-th token in the i-th candidate document, respectively. And let Q_m denote the query embedding for the m-th token in the decoding phase of LLM, the standard attention computation is defined as:

$$\mathcal{A}(Q_m, K_{i,t}) = \frac{\exp\left(\frac{Q_m K_{i,t}^T}{\sqrt{d_k}}\right)}{\sum_{i'} \sum_{t'} \exp\left(\frac{Q_m K_{i',t'}^T}{\sqrt{d_k}}\right)} \tag{2}$$

where $\mathcal{A}(Q_m, K_{i,t})$ represents the attention score of the m-th token in the decoding process to the t-th token in the i-th candidate document.

The reranker retains only the top-k documents for attention computation, and these top-k documents, as a set, are used as part of the prompt. The order of these documents is no longer important. Therefore, we can use a corresponding *hard* attention mask $\mathcal{M}^{\mathcal{R}}$ to simulate the reranking process.

$$\mathcal{M}_i^{\mathcal{R}} = \begin{cases} 1, & \text{if } i \in \mathcal{I}_k \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

$$\mathcal{MA}(Q_m, K_{i,t}) = \frac{\mathcal{M}_i^{\mathcal{R}} \exp\left(\frac{Q_m K_{i,t}^T}{\sqrt{d_k}}\right)}{\sum_{i'} \mathcal{M}_{i'}^{\mathcal{R}} \sum_{t'} \exp\left(\frac{Q_m K_{i',t'}^T}{\sqrt{d_k}}\right)} \tag{4}$$

This formulation of masked attention is mathematically equivalent to reranking. If document i is not selected by the reranker, i.e., $\mathcal{M}_i^{\mathcal{R}} = 0$, then all tokens within document i receive an attention score of zero, i.e., $\mathcal{MA}(Q_m, K_{i,t}) = 0, \ \forall t$.

Algorithm 1 Gumbel Reranking: Training Reranker via Differentiable Masked Attention

procedure StochasticSubsetMask(reranker $\mathcal{R}$, documents $\mathbf{d_1}, \ldots, \mathbf{d_n}$, query $\mathbf{q}$, temperature τ, scale factor κ, subset size k)
    $w_i = \mathcal{R}(\text{Concatenate}(\mathbf{q}, \mathbf{d}_i)) \quad \forall i \in [n] \triangleq \{1, 2, \ldots, n\}$ ▷ Apply Reranker
    for $j = 1$ **to** k **do** ▷ Stochastic Top-k Sampling
        $\tilde{w}_i = -\log(-\log(u_i)) + \kappa \cdot w_i, \quad u_i \sim \mathcal{U}(0,1) \quad \forall i \in [n]$
        $\hat{\mathcal{M}}^{\mathcal{R},j} = \text{softmax}\left(\frac{\tilde{\mathbf{w}}}{\tau}\right), \quad \tilde{\mathbf{w}} = (\tilde{w}_1, \tilde{w}_2, \ldots, \tilde{w}_n)$
    end for
    return $\max(\hat{\mathcal{M}}^{\mathcal{R},1}, \ldots, \hat{\mathcal{M}}^{\mathcal{R},k})$ ▷ Return Relaxed Top-k Mask
end procedure

for each (query $\mathbf{q}$, answer $\mathbf{a}$) in training data **do** ▷ Training Loop
    Retrieve n documents $\mathbf{d_1}, \ldots, \mathbf{d_n}$ using $\mathbf{q}$
    $\hat{\mathcal{M}}^{\mathcal{R}} = \text{StochasticSubsetMask}(\mathcal{R}, \mathbf{d_1}, \ldots, \mathbf{d_n}, \mathbf{q}, \tau, \kappa, k)$
    Apply $\mathcal{DMA}(\hat{\mathcal{M}}^{\mathcal{R}})$ to obtain logits and language loss $\mathcal{L}_{LM}$ ▷ subsection 3.3
    Update reranker $\mathcal{R}$ with $\nabla_{\mathcal{R}} \mathcal{L}_{LM}$ ▷ Reranker Optimization
end for

Independence Requirements in Pre-Filling To effectively simulate reranking via $\mathcal{MA}$, it is crucial to ensure the independence of candidate documents. First, all candidate documents should use the same positional encoding to eliminate position bias. Second, each document should be encoded independently during pre-filling to prevent information leakage across documents. To enforce these constraints, we adopt the parallel pre-filling architecture, as seen in models like FiD (Izacard and Grave, 2021b), CEPE (Yen et al., 2024), and Parallel Windows (Ratner et al., 2023), where retrieved documents are encoded separately with independent position encodings during the pre-filling stage.

3.3 Differentiable Masked Attention

The problem of learning a more effective reranker is thus reformulated as learning a better attention mask $\mathcal{M}^{\mathcal{R}}$. However, the *hard* attention mask $\mathcal{M}^{\mathcal{R}}$ defined in Equation 3 remains non-differentiable, preventing end-to-end optimization based on the final language loss. To solve this problem, we leverage the Gumbel-Softmax technique (Jang et al., 2017) to convert discrete sampling into a differentiable process. Specifically, we transform the reranker's output $w_1, w_2, \ldots, w_n$ into a probability distribution for sampling an attention mask:

$$G_i = -\log\Big(-\log(u_i)\Big), \quad u_i \sim \mathcal{U}(0,1),$$
$$\hat{\mathcal{M}}^{\mathcal{R}} = \frac{\exp\Big(\frac{\tilde{w}_i}{\tau}\Big)}{\sum_{j=1}^{n} \exp\Big(\frac{\tilde{w}_j}{\tau}\Big)}, \quad \tilde{w}_i = G_i + \kappa \cdot w_i \tag{5}$$

where $\hat{\mathcal{M}}_i^{\mathcal{R}}$ represents the probability of selecting the i-th document. τ and κ are hyperparameters in the Gumbel training process. We discuss their effects in detail in Appendix A. To approximate Top-k reranking, we perform independent sampling k times and compute the element-wise maximum:

$$\hat{\mathcal{M}}^{\mathcal{R}} = \max\Big(\hat{\mathcal{M}}^{\mathcal{R},1}, \ldots, \hat{\mathcal{M}}^{\mathcal{R},k}\Big) \tag{6}$$

This results in a *soft* attention mask representing the sampled subset, leading to Differentiable Masked Attention:

$$\mathcal{DMA}(Q_m, K_{i,t}) = \frac{\hat{\mathcal{M}}_i^{\mathcal{R}} \exp\left(\frac{Q_m K_{i,t}^T}{\sqrt{d_k}}\right)}{\sum_{i'} \hat{\mathcal{M}}_{i'}^{\mathcal{R}} \sum_{t'} \exp\left(\frac{Q_m K_{i',t'}^T}{\sqrt{d_k}}\right)} \tag{7}$$

This formulation allows end-to-end optimization of the reranker $\mathcal{R}$ based on final language model loss, improving overall RAG system performance.

3.4 Gumbel Reranking Pipeline

In this section, we introduce *Gumbel Reranking*, an end-to-end reranker optimization framework leveraging the previously introduced $\mathcal{DMA}$. The overall pipeline is outlined in Algorithm 1.

Training Process Given a query $\mathbf{q}$ and a set of candidate documents $\mathbf{d_1}, \ldots, \mathbf{d_n}$, the reranker first computes a relevance score for each document. The Stochastic Subset Mask algorithm then generates a Top-k attention mask $\hat{\mathcal{M}}^{\mathcal{R},k}$, which represents the probability of selecting each candidate document.

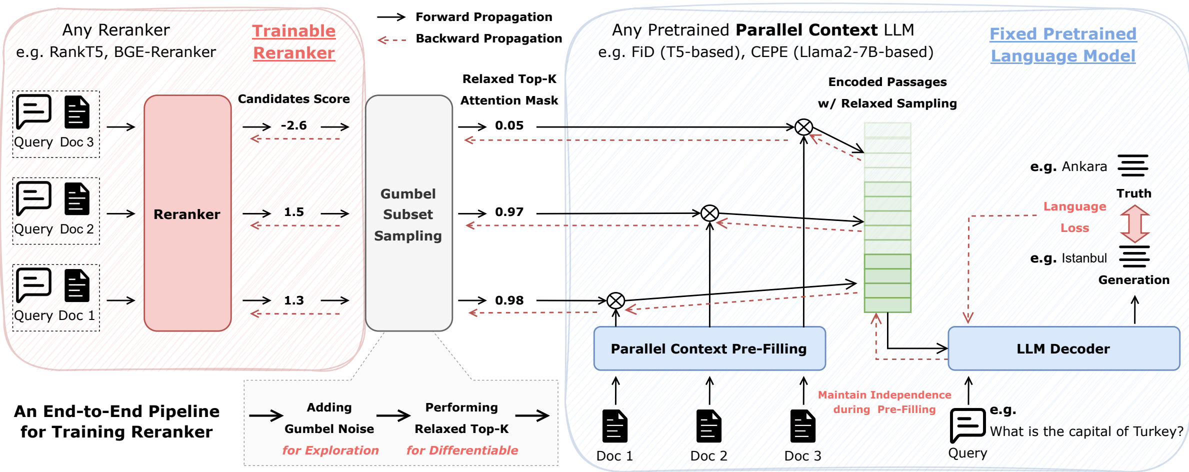

Figure 2: Implementation workflow of G-Rerank. We focus on fine-tuning the reranker while keeping LLM parameters fixed. However, given sufficient computational resources, joint fine-tuning of both the LLM and the reranker is feasible. In the Pre-Filling phase, it is essential to maintain the independence of candidate documents.

The selected documents are subsequently used in the generation process, where the attention mechanism follows Equation 7 to compute logits and the language modeling loss. Finally, the reranker is optimized by minimizing the language loss $\mathcal{L}_{LM}$. Since our proposed framework primarily focuses on enhancing the reranking module, we fix the parameters of the LLM in our experimental setup, as shown in Figure 2. This also facilitates a fairer comparison of different reranker training strategies.

Key Advantages This framework facilitates end-to-end optimization of the reranker via backpropagation, offering two primary advantages. First, by modeling reranking as applying a document-wise attention mask, it mitigates the discrepancy between training and inference, guiding the reranker to prioritize documents that minimize the final generation loss. Second, our approach leverages gumbel subset sampling, enabling the model to identify the complete evidence *subset* during training, rather than analyzing each candidate document independently. This advantage makes our method well-suited for multi-hop QA scenarios and sets it apart from existing perplexity-based distillation techniques, as discussed in Appendix D.

4 Experiments

In subsection 4.2, we first validate the effectiveness of our approach under three different experimental settings. Then, in subsection 4.3, we focus on whether the reranker can learn to prioritize *indirect evidence* in multi-hop question answering. Next, in subsection 4.4, we conduct an ablation study on the Gumbel trick and demonstrate its necessity. Finally, in subsection 4.5, we remove the reranker and assign each document a learnable weight to further verify the efficacy of our training objective in capturing the relative importance of documents.

4.1 Experimental Setup

Language Models We experiment with two different language models as the generation module in our RAG system: Fusion-in-Decoder (FiD) (Izacard and Grave, 2021b) and CEPE-Llama2-7B (Yen et al., 2024). FiD (Raffel et al., 2020), built upon the T5 architecture, is specifically designed for knowledge-intensive QA and is fine-tuned for each task. CEPE-Llama2-7B segments long documents with a lightweight encoder and employs cross-attention for effective context utilization, operating in a zero-shot manner.

Reranker We experiment with RankT5-Base (Zhuang et al., 2023) and BGE-Base-Reranker (Xiao and Liu, 2023) as the reranking module in the RAG system. RankT5-Base is fine-tuned in an encoder-decoder setup to perform reranking, while BGE-Base-Reranker is an encoder-only model based on BERT.

Datasets We evaluate on five QA datasets: multi-hop (2WikiHop (Ho et al., 2020), HotpotQA (Yang et al., 2018), Musique (Trivedi et al., 2022)) and single-hop (NQ (Kwiatkowski et al., 2019), TQA (Kim et al., 2019)). Details are in subsection C.3. For NQ and TQA, we retrieve 20 candidate documents per query using DPR (Karpukhin et al., 2020). For multi-hop datasets, we apply

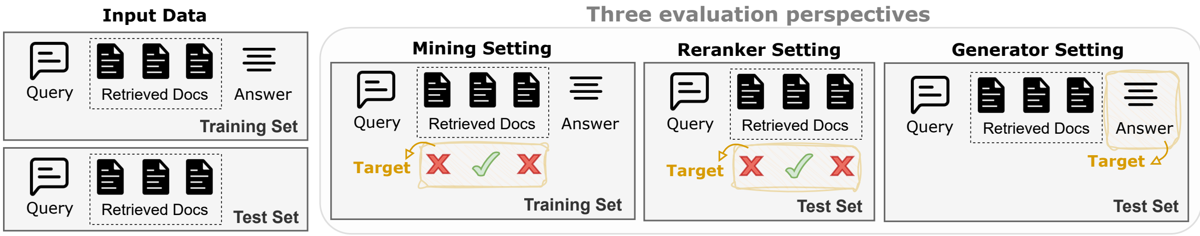

Figure 3: Comparison of three different experimental settings. In addition to common evaluation metrics on the test set, we also assess the reranker's ability to identify relevant documents from the training set.

Training Methods	Mining Setting: Recall@5	Mining Setting: NDCG@5	Reranker Setting: Recall@5	Reranker Setting: NDCG@5	Reranker Setting: MRR	Generator Setting: EM	Generator Setting: SubEM	Generator Setting: F1
Dataset: `Hotpotqa`								
Reranker: **`RankT5`**								
- EMDR (Lin et al., 2024)	78.0	80.5	78.7	80.6	**95.9**	60.8	66.1	75.8
- PDist (Glass et al., 2022)	76.8	79.5	78.1	80.0	95.7	60.8	66.0	75.8
- LOOP (Izacard et al., 2023)	71.7	74.7	72.5	74.9	93.0	60.0	65.1	75.0
- ADist (Izacard and Grave, 2021a)	71.3	72.1	71.3	71.9	88.4	57.0	61.9	71.5
- G-Rerank	**83.3**	**84.7**	**84.4**	**84.9**	**95.9**	**61.1**	**66.5**	**76.3**
Reranker: **`BGE-Base`**								
- EMDR (Lin et al., 2024)	81.1	83.2	**81.8**	**83.1**	**96.3**	60.8	66.0	**75.8**
- PDist (Glass et al., 2022)	79.1	81.6	81.2	82.6	96.2	**60.9**	**66.1**	75.7
- LOOP (Izacard et al., 2023)	79.1	81.1	80.4	81.7	95.3	60.3	65.4	75.2
- ADist (Izacard and Grave, 2021a)	77.7	79.5	78.1	79.5	93.7	59.8	65.0	74.7
- G-Rerank	**81.6**	**83.3**	81.1	82.9	95.8	**60.9**	**66.1**	75.7
Dataset: `Musique`								
Reranker: **`RankT5`**								
- EMDR (Lin et al., 2024)	56.6	65.8	55.0	58.1	**82.0**	39.6	42.1	48.6
- PDist (Glass et al., 2022)	57.3	65.3	52.7	55.0	79.5	39.6	42.2	48.3
- LOOP (Izacard et al., 2023)	56.3	64.9	53.3	55.6	79.6	39.2	41.7	48.0
- ADist (Izacard and Grave, 2021a)	53.8	55.3	47.7	47.3	66.4	35.4	37.9	44.1
- G-Rerank	**60.7**	**67.8**	**57.9**	**59.7**	81.5	**40.0**	**42.4**	**49.1**
Reranker: **`BGE-Base`**								
- EMDR (Lin et al., 2024)	56.6	65.7	53.6	57.1	81.5	39.7	42.4	48.8
- PDist (Glass et al., 2022)	60.3	66.1	**58.2**	59.6	80.5	39.4	42.3	48.6
- LOOP (Izacard et al., 2023)	58.7	65.6	57.2	59.3	**81.8**	39.7	42.2	48.8
- ADist (Izacard and Grave, 2021a)	57.9	64.5	46.0	45.3	64.7	34.8	37.3	43.4
- G-Rerank	**60.9**	**66.6**	57.6	**59.7**	81.5	**39.9**	**42.7**	**49.1**
Dataset: `2wikihop`								
Reranker: **`RankT5`**								
- EMDR (Lin et al., 2024)	58.6	63.4	62.9	68.7	88.7	67.2	69.9	72.5
- PDist (Glass et al., 2022)	72.6	76.5	77.2	81.9	94.1	70.2	73.0	75.5
- LOOP (Izacard et al., 2023)	80.4	**87.1**	79.2	85.4	97.5	71.6	74.4	76.9
- ADist (Izacard and Grave, 2021a)	74.7	79.2	72.4	76.6	90.1	64.1	66.5	69.6
- G-Rerank	**80.8**	86.9	**82.7**	**88.4**	**97.8**	**71.8**	**74.7**	**77.2**
Reranker: **`BGE-Base`**								
- EMDR (Lin et al., 2024)	61.8	67.3	71.0	77.1	93.8	68.9	71.8	74.3
- PDist (Glass et al., 2022)	74.0	76.8	76.6	82.2	94.5	69.1	71.9	74.4
- LOOP (Izacard et al., 2023)	77.3	85.0	76.0	83.3	**98.5**	**71.2**	**73.9**	**76.3**
- ADist (Izacard and Grave, 2021a)	**81.4**	**87.7**	80.5	86.4	97.1	70.7	73.5	76.1
- G-Rerank	79.6	86.4	**81.4**	**86.5**	97.5	70.9	73.7	76.2

Table 1: Experiments on 2WikiHop, Musique, and HotpotQA using FiD-Large as reader. We consider the settings illustrated in Figure 3. The best performance is highlighted in bold, while the second-best performance is underlined.

the distraction setting to ensure ground-truth documents are included, adding 10 random candidates in Musique to maintain 20 candidates per query.

Baselines We compare against four LLM-supervised reranker training methods: EMDR (Sachan et al., 2021; Shi et al., 2024; Lin et al., 2024), PDist (Izacard et al., 2023; Glass et al., 2022), LOOP (Izacard et al., 2023), and ADist (Izacard and Grave, 2021a), which employ different LLM-supervised losses. Details about these baselines can be found in Appendix B.

4.2 Main Experiments

Task Definition We consider the QA task where the model is trained on question-answer pairs along with retrieved documents, but at test time, it only receives the question and the retrieved documents. We define three evaluation settings, with their respective distinctions illustrated in Figure 3:

1. **Mining Setting**: During training, given a *question-answer pair*, can the reranker effectively identify relevant documents?

	Reranker Setting						Mining Setting					
	FiD-Base			FiD-Large			FiD-Base			FiD-Large		
Training Methods	Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG
- EMDR (Lin et al., 2024)	63.0	45.6	45.2	61.8	45.2	44.4	60.3	42.9	42.3	59.0	42.7	41.6
- PDist (Glass et al., 2022)	50.5	39.8	36.2	60.2	44.4	43.4	47.6	37.5	33.5	56.3	41.3	39.7
- LOOP (Izacard et al., 2023)	53.1	40.7	38.1	52.5	40.2	37.3	49.6	38.2	34.9	49.8	37.6	34.5
- ADist (Izacard and Grave, 2021a)	55.2	43.4	40.8	56.3	44.5	41.9	52.8	41.5	38.5	54.2	42.3	39.5
- G-Rerank	**69.3**	**48.2**	**49.6**	**72.2**	**49.5**	**51.5**	**65.5**	**45.0**	**45.8**	**68.4**	**46.4**	**47.8**

Table 2: Results on HotpotQA using FiD as reader for identifying *indirectly relevant documents*, which are part of the evidence chain but do not directly contain the answer. Details can be found in Appendix E.

	RankT5		BGE-Base	
Training Methods	NQ	TQA	NQ	TQA
- EMDR (Lin et al., 2024)	33.4	62.4	33.7	62.5
- PDist (Glass et al., 2022)	32.9	61.8	33.9	61.7
- LOOP (Izacard et al., 2023)	33.7	62.1	33.5	62.2
- ADist (Izacard and Grave, 2021a)	33.1	61.6	33.2	62.0
- G-Rerank	**34.3**	**62.8**	**34.5**	**63.1**

Table 3: Experimental results on NQ and TQA datasets using CEPE-Llama2-7B as the reader. We employ SubEM as the evaluation metric.

2. **Reranker Setting**: At test time, given a *question*, can the reranker effectively identify relevant documents?

3. **Generator Setting**: At test time, given a *question*, can the model generate correct answers?

Experimental Results Table 1 presents the experimental results using FiD-Large as the generator model. Our method, G-Rerank, achieves the best or second-best performance across most datasets. In the Mining Setting, G-Rerank significantly improves the ability to identify relevant documents during training, given question-answer pairs. For instance, it achieves a 5.3% improvement on the `HotpotQA` when using RankT5. In the Reranker Setting, G-Rerank demonstrates a notable improvement over other LLM-supervised loss-based training methods, with a 5.7% Recall improvement on `HotpotQA` when using RankT5. Furthermore, in the Generator Setting, G-Rerank shows consistent performance gains in generation quality, as G-Rerank directly takes the minimization of the final generation loss as the training objective.

Table 3 presents the SubEM results using CEPE-Llama2-7B as the generator model. We do not fine-tune CEPE-Llama2-7B on the downstream datasets; instead, we leverage its zero-shot capabilities. On both `NQ` and `TQA`, the G-Rerank training strategy leads to the best generation performance. Notably, these improvements are achieved solely by fine-tuning the retrieval module while keeping the language model parameters fixed.

4.3 Identifying Indirectly Relevant Documents

In multi-hop question answering, a RAG system is required to retrieve a complete evidence chain comprising multiple documents to support its final answer. In such scenarios, the reranker should be able to identify *indirectly relevant documents*, which are relevant to the query but do not directly contain the final answer. The challenge, however, lies in the fact that these documents often serve as 'partial' evidence, and their relevance is not immediately apparent without being combined with other documents. Existing perplexity-based training methods commonly used in the literature distill independent relevance scores for each document, which fail to capture the inter-document dependencies that are essential for identifying indirectly relevant documents, as discussed in Appendix D.

We evaluate various reranker training methods on `HotpotQA` to assess their ability to identify *indirectly relevant documents*. To obtain the data, we employ a straightforward rule-based method to extract such documents: any document labeled as relevant in the dataset but not directly containing the final answer is considered an indirectly relevant document. Further discussion about this rule can be found in Appendix E.

The experimental results are summarized in Table 2. Our method, G-Rerank, demonstrates a significant improvement in identifying indirectly relevant documents. Specifically, when FiD-Large is used as the generator model, G-Rerank achieves a recall improvement of 10.4%. These results suggest that our approach, which views reranking as a subset sampling problem, allows the model to better capture inter-document relationships and effectively recognize complete evidence chains.

4.4 Necessity of Gumbel Trick

We leverage the Gumbel trick to transform the output weights of the reranker into an approximately

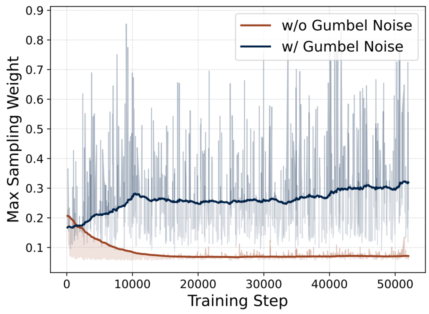

Figure 4: Comparison of Max Sampling Weight (indicating the reranker's ability to distinguish between candidate documents) with and without Gumbel noise on the NQ dataset.

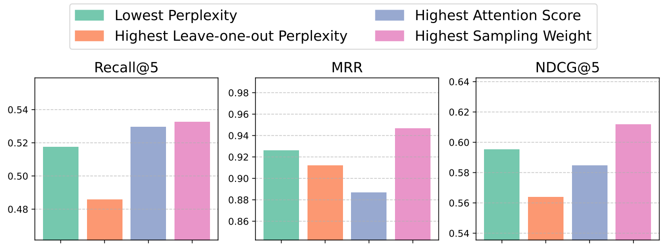

Figure 5: Performance comparison of different scalar metrics for assessing candidate document relevance in the Mining Setting. Our method is illustrated in Figure 8 and Algorithm 2, while other baseline methods are described in detail in Appendix F.

discrete attention mask, where values tend to converge to either 0 or 1. A natural question arises: *Is the introduction of Gumbel noise essential?* We conduct an ablation study by removing the Gumbel noise and directly utilizing the reranker's output weights as the attention mask while maintaining the same end-to-end optimization process.

Our experiments reveal a substantial drop in performance when Gumbel noise is omitted. Specifically, the EM metric on the NQ dataset decreases drastically from 46.2 (with Gumbel) to 12.7 (without Gumbel). To gain further insight, we visualize the reranker's output weights during training.

Figure 4 presents the average maximum normalized document weight assigned by the reranker. With the Gumbel noise applied, we observe a clear upward trend in the maximum document weight, indicating that the reranker progressively enhances the differentiation between candidate documents, which ultimately leads to convergence. In contrast, when Gumbel noise is removed, the maximum document weight decreases over time, eventually stabilizing at 0.05, signaling a diminished ability to distinguish between candidates. This degradation occurs because, in the absence of the discretization constraint introduced by the Gumbel trick, the model tends to preserve the original attention distribution, thus treating the removal of the attention mask as its objective. Consequently, the reranker learns to assign uniform soft mask across all candidates, i.e., $\hat{\mathcal{M}}^{\mathcal{R}}_{i,\text{w/o Gumbel}} = \frac{1}{N} = 0.05, \forall i$, thereby reverting the masked attention mechanism Equation 7 to its original form as defined in Equation 2. These findings underscore the importance of the discretization constraint imposed by the Gumbel trick for learning an effective attention mask.

4.5 Learnable Sampling Weights

The presence of the reranker can be viewed as incorporating text-based prior knowledge into the document relevance learning process. However, even in the absence of text-based priors, our training methodology can still effectively identify the relevant documents. To verify this, we focus on the Mining Setting and investigate whether the model is capable of learning meaningful document relevance scores *without the use of a reranker*.

Our method is illustrated in Figure 8 and Algorithm 2, while the experimental setup and baseline methods are explained in Appendix F. Specifically, we remove the reranker component and instead assign each candidate document a learnable sampling weight, initializing all weights to zero. The results, presented in Figure 5, show that even without the reranker (i.e., without prior knowledge of the text), our approach is still able to learn reliable relevance scores for each document. Moreover, it significantly outperforms other scalar metrics based on perplexity or attention scores, further confirming the effectiveness of our training objective.

5 Conclusion

In this work, we introduce G-Rerank, an end-to-end optimization framework for training rerankers in RAG systems. By reinterpreting the reranking process as masked attention, we leverage the Gumbel Trick and Relaxed Top-k to enable direct optimization of the document-wise attention mask. Our method effectively captures document interdependencies and aligns retrieval and generation objectives. Experiments across different settings show that G-Rerank notably improves reranker performance, especially in multi-hop QA tasks.

6 Limitations

Our method imposes certain constraints on its applicability to existing decoder-only large language models (LLMs) due to its reliance on parallel encoding/decoding capabilities during the pre-filling stage. This requirement limits its direct adoption in conventional autoregressive LLMs. However, it is worth noting that many high-performance language models with parallel encoding/decoding capabilities have already become standard choices in various Retrieval-Augmented Generation (RAG) systems, such as FiD (Izacard and Grave, 2021b), CEPE (Yen et al., 2024), and Parallel Windows (Ratner et al., 2023). Furthermore, our approach requires such models only during the reranker training phase; once trained, the reranker itself is independent of any specific LLM and can be flexibly adapted to other decoder-only models. Therefore, our method primarily serves as a general training framework rather than imposing architectural constraints on the final inference model. Additionally, our approach introduces extra hyperparameters in the Gumbel-Softmax process, including the temperature parameter τ and the scaling factor κ, which require tuning to achieve optimal performance. However, through empirical studies, we find that $\tau = 0.5$ and $\kappa = 1.0$ provide robust and stable performance across different model architectures and datasets. We provide a further discussion on the effect of τ and κ in Appendix A.

7 Ethical Considerations

While our method aims to improve the accuracy of the RAG system, it does not eliminate the inherent risks of biased data or model outputs, as the performance of RAG systems still heavily depends on the quality of training data and underlying models. The potential for bias in the training data, particularly for domain-specific queries, can lead to the amplification of these biases in the retrieved results, which can impact downstream applications.

References

Akari Asai, Matt Gardner, and Hannaneh Hajishirzi. 2022. Evidentiality-guided generation for knowledge-intensive NLP tasks. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 2226–2243. Association for Computational Linguistics.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George 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 W. Rae, Erich Elsen, and Laurent Sifre. 2022. Improving language models by retrieving from trillions of tokens. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 2206–2240. PMLR.

Jianbo Chen, Le Song, Martin J. Wainwright, and Michael I. Jordan. 2018. Learning to explain: An information-theoretic perspective on model interpretation. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 882–891. PMLR.

Eunseong Choi, Hyeri Lee, and Jongwuk Lee. 2024. Multi-granularity guided fusion-in-decoder. In *Findings of the Association for Computational Linguistics: NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 2201–2212. Association for Computational Linguistics.

Michiel de Jong, Yury Zemlyanskiy, Joshua Ainslie, Nicholas FitzGerald, Sumit Sanghai, Fei Sha, and William W. Cohen. 2023. Fido: Fusion-in-decoder optimized for stronger performance and faster inference. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 11534–11547. Association for Computational Linguistics.

Jialin Dong, Bahare Fatemi, Bryan Perozzi, Lin F. Yang, and Anton Tsitsulin. 2024. Don't forget to connect! improving RAG with graph-based reranking. *CoRR*, abs/2405.18414.

Zhihao Fan, Yeyun Gong, Dayiheng Liu, Zhongyu Wei, Siyuan Wang, Jian Jiao, Nan Duan, Ruofei Zhang, and Xuanjing Huang. 2021. Mask attention networks: Rethinking and strengthen transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 1692–1701. Association for Computational Linguistics.

Gongfan Fang, Hongxu Yin, Saurav Muralidharan, Greg Heinrich, Jeff Pool, Jan Kautz, Pavlo Molchanov, and Xinchao Wang. 2024. Maskllm: Learnable semi-structured sparsity for large language models. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Michael R. Glass, Gaetano Rossiello, Md. Faisal Mahbub Chowdhury, Ankita Naik, Pengshan Cai, and Alfio Gliozzo. 2022. Re2g: Retrieve, rerank, generate. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 2701–2715. Association for Computational Linguistics.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Retrieval augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938. PMLR.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing A multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 6609–6625. International Committee on Computational Linguistics.

Sebastian Hofstätter, Jiecao Chen, Karthik Raman, and Hamed Zamani. 2023. Fid-light: Efficient and effective retrieval-augmented text generation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, pages 1437–1447. ACM.

Gautier Izacard and Edouard Grave. 2021a. Distilling knowledge from reader to retriever for question answering. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Gautier Izacard and Edouard Grave. 2021b. 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, EACL 2021, Online, April 19 - 23, 2021*, pages 874–880. Association for Computational Linguistics.

Gautier Izacard, Patrick S. H. Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: Few-shot learning with retrieval augmented language models. *J. Mach. Learn. Res.*, 24:251:1–251:43.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

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

Zixuan Ke, Weize Kong, Cheng Li, Mingyang Zhang, Qiaozhu Mei, and Michael Bendersky. 2024. Bridging the preference gap between retrievers and llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 10438–10451. Association for Computational Linguistics.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 39–48. ACM.

Daesik Kim, Seonhoon Kim, and Nojun Kwak. 2019. Textbook question answering with multi-modal context graph understanding and self-supervised open-set comprehension. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3568–3584. Association for Computational Linguistics.

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

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 6086–6096. Association for Computational Linguistics.

Patrick S. H. 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. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Xin-Yi Li, Wei-Jun Lei, and Yu-Bin Yang. 2023. From easy to hard: Two-stage selector and reader for multi-hop question answering. In *IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2023, Rhodes Island, Greece, June 4-10, 2023*, pages 1–5. IEEE.

Zhenyu Li, Yike Zhang, Tengyu Pan, Yutao Sun, Zhichao Duan, Junjie Fang, Rong Han, Zixuan Wang, and Jianyong Wang. 2024. Focusllm: Scaling llm's context by parallel decoding. *CoRR*, abs/2408.11745.

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. 2024. RA-DIT: retrieval-augmented dual instruction tuning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Nir Ratner, Yoav Levine, Yonatan Belinkov, Ori Ram, Inbal Magar, Omri Abend, Ehud Karpas, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. Parallel context windows for large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 6383–6402. Association for Computational Linguistics.

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

Devendra Singh Sachan, Mike Lewis, Dani Yogatama, Luke Zettlemoyer, Joelle Pineau, and Manzil Zaheer. 2023. Questions are all you need to train a dense passage retriever. *Trans. Assoc. Comput. Linguistics*, 11:600–616.

Devendra Singh Sachan, Siva Reddy, William L. Hamilton, Chris Dyer, and Dani Yogatama. 2021. End-to-end training of multi-document reader and retriever for open-domain question answering. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 25968–25981.

Alireza Salemi and Hamed Zamani. 2024. Evaluating retrieval quality in retrieval-augmented generation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024*, pages 2395–2400. ACM.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Richard James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2024. REPLUG: retrieval-augmented black-box language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 8371–8384. Association for Computational Linguistics.

Yixuan Tang and Yi Yang. 2024. Multihop-rag: Benchmarking retrieval-augmented generation for multi-hop queries. *CoRR*, abs/2401.15391.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Musique: Multi-hop questions via single-hop question composition. *Trans. Assoc. Comput. Linguistics*, 10:539–554.

Ming Tu, Kevin Huang, Guangtao Wang, Jing Huang, Xiaodong He, and Bowen Zhou. 2020. Select, answer and explain: Interpretable multi-hop reading comprehension over multiple documents. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9073–9080. AAAI Press.

Cunxiang Wang, Haofei Yu, and Yue Zhang. 2023. Rfid: Towards rational fusion-in-decoder for open-domain question answering. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 2473–2481. Association for Computational Linguistics.

Bohong Wu, Zhuosheng Zhang, and Hai Zhao. 2021. Graph-free multi-hop reading comprehension: A select-to-guide strategy. *CoRR*, abs/2107.11823.

Shitao Xiao and Zheng Liu. 2023. Baai general embedding.

Sang Michael Xie and Stefano Ermon. 2019. Reparameterizable subset sampling via continuous relaxations. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 3919–3925. ijcai.org.

Yujia Xie, Hanjun Dai, Minshuo Chen, Bo Dai, Tuo Zhao, Hongyuan Zha, Wei Wei, and Tomas Pfister. 2020. Differentiable top-k with optimal transport. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Wenhan Xiong, Xiang Lorraine Li, Srini Iyer, Jingfei Du, Patrick S. H. Lewis, William Yang Wang, Yashar Mehdad, Scott Yih, Sebastian Riedel, Douwe Kiela, and Barlas Oguz. 2021. Answering complex open-domain questions with multi-hop dense retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Jiancheng Yang, Qiang Zhang, Bingbing Ni, Linguo Li, Jinxian Liu, Mengdie Zhou, and Qi Tian. 2019.

Modeling point clouds with self-attention and gumbel subset sampling. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 3323–3332. Computer Vision Foundation / IEEE.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2369–2380. Association for Computational Linguistics.

Howard Yen, Tianyu Gao, and Danqi Chen. 2024. Long-context language modeling with parallel context encoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 2588–2610. Association for Computational Linguistics.

Zhangyue Yin, Yuxin Wang, Xiannian Hu, Yiguang Wu, Hang Yan, Xinyu Zhang, Zhao Cao, Xuanjing Huang, and Xipeng Qiu. 2023. Rethinking label smoothing on multi-hop question answering. In *Chinese Computational Linguistics - 22nd China National Conference, CCL 2023, Harbin, China, August 3-5, 2023, Proceedings*, volume 14232 of *Lecture Notes in Computer Science*, pages 72–87. Springer.

Donghan Yu, Chenguang Zhu, Yuwei Fang, Wenhao Yu, Shuohang Wang, Yichong Xu, Xiang Ren, Yiming Yang, and Michael Zeng. 2022. Kg-fid: Infusing knowledge graph in fusion-in-decoder for open-domain question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 4961–4974. Association for Computational Linguistics.

Jiahao Zhang, Haiyang Zhang, Dongmei Zhang, Yong Liu, and Shen Huang. 2024. End-to-end beam retrieval for multi-hop question answering. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 1718–1731. Association for Computational Linguistics.

Chen Zhao, Chenyan Xiong, Jordan L. Boyd-Graber, and Hal Daumé III. 2021. Multi-step reasoning over unstructured text with beam dense retrieval. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 4635–4641. Association for Computational Linguistics.

Fengbin Zhu, Wenqiang Lei, Chao Wang, Jianming Zheng, Soujanya Poria, and Tat-Seng Chua. 2021. Retrieving and reading: A comprehensive survey on open-domain question answering. *CoRR*, abs/2101.00774.

Honglei Zhuang, Zhen Qin, Rolf Jagerman, Kai Hui, Ji Ma, Jing Lu, Jianmo Ni, Xuanhui Wang, and Michael Bendersky. 2023. Rankt5: Fine-tuning T5 for text ranking with ranking losses. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, pages 2308–2313. ACM.

A Effect of hyper-parameters on the Training Process

The temperature parameter τ controls the sharpness of the softmax distribution used in the selection process of documents. We conduct sampling weight learning for 20 candidate documents based on RankT5 on the NQ dataset, and tested the impact of different τ values on the sampling weights during the training process. The experimental results are shown in Figure 6. Specifically, as τ approaches zero, the softmax distribution becomes increasingly sharp, leading to a *hard* selection process where the model heavily favors the document with the highest score. This results in a deterministic decision-making process, where the model's focus is on exploitation, quickly converging to a particular document. On the other hand, when τ increases, the distribution becomes smoother, allowing for a more stochastic sampling process. This introduces more exploration, as the model is less likely to fixate on a single document, encouraging the exploration of other potential candidates. A larger τ thus promotes diversity in the selection process, which can be beneficial for avoiding local optima and improving generalization during training.

The scaling factor κ plays a critical role in controlling the relative influence of the Reranker scores on the overall document selection process. We test the impact of different κ values on the sampling weights during the training process. The experimental results are shown in Figure 7. Specifically, κ modulates the contribution of the Reranker score w_i to the final selection probability. When κ is small, the contribution of the original Gumbel noise term G_i dominates the selection process. This introduces significant randomness, increasing the exploration rate during training. A small κ value results in noisy selection, encouraging the model to explore various documents and learn more diverse representations. Conversely, when κ is large, the Reranker score w_i has a stronger influence, and the model's selection becomes more deterministic. In this case, the Reranker score dominates the sampling process, leading to faster convergence as the model focuses on selecting the most highly scored documents. However, an overly large κ may limit the exploration of alternative options, potentially leading to overfitting and reduced generalization.

B LLM-Supervised Baselines

Baselines. We compare our approach against four LLM-supervised reranker training methods that leverage generative language model signals to supervise retriever learning without requiring additional document annotations. In particular, we consider the following methods:

Attention Distillation (ADist) (Izacard and Grave, 2021a): This method utilizes the cross-attention scores from the language model—augmented by the norms of the corresponding value vectors—to compute a target relevance distribution over retrieved documents. The reranker $\mathcal{R}$ is trained by minimizing the KL-divergence between its own distribution over the top-k documents and the attention-based target distribution. The target distribution for the reranker is defined as:

$$p_{\text{ATTN}}(\mathbf{p}_k) = \frac{\exp(\alpha_k \|\mathbf{v}_k\|_2)}{\sum_{i=1}^{K} \exp(\alpha_i \|\mathbf{v}_i\|_2)}$$

where α_k is the attention score for document $\mathbf{p}_k$ and $\|\mathbf{v}_k\|_2$ is the L2 norm of the corresponding value vector. The loss function minimizes the KL-divergence between the reranker's distribution $p_{\mathcal{R}}$ and the target distribution p_{ATTN}:

$$\text{KL}(p_{\text{ATTN}} \parallel p_{\mathcal{R}}) = \sum_{k=1}^{K} p_{\text{ATTN}}(\mathbf{p}_k) \log \frac{p_{\text{ATTN}}(\mathbf{p}_k)}{p_{\mathcal{R}}(\mathbf{p}_k)}$$

End-to-end Multi-Document Reader and Reranker (EMDR^2) (Sachan et al., 2021; Shi et al., 2024; Lin et al., 2024): EMDR^2 adopts an expectation-maximization approach, treating the retrieved documents as latent variables. Given a query $\mathbf{q}$ and a corresponding answer $\mathbf{a}$, along with the top-k retrieved documents, the loss is designed to maximize the log-likelihood of the output given these documents. The objective function is:

$$\mathcal{L}_{\text{EMDR}^2} = \log \left[\sum_{k=1}^{K} p_{\text{LM}}(\mathbf{a} \mid \mathbf{q}, \mathbf{p}_k) p_{\mathcal{R}}(\mathbf{p}_k \mid \mathbf{q}) \right]$$

where $p_{\text{LM}}(\mathbf{a} \mid \mathbf{q}, \mathbf{p}_k)$ is the language model's probability of generating the answer $\mathbf{a}$ conditioned on the query $\mathbf{q}$ and document $\mathbf{p}_k$, and $p_{\mathcal{R}}(\mathbf{p}_k \mid \mathbf{q})$ is the reranker's distribution over the top-k documents.

Perplexity Distillation (PDist) (Izacard et al., 2023; Glass et al., 2022): In this approach, the reranker is trained to predict the improvement in

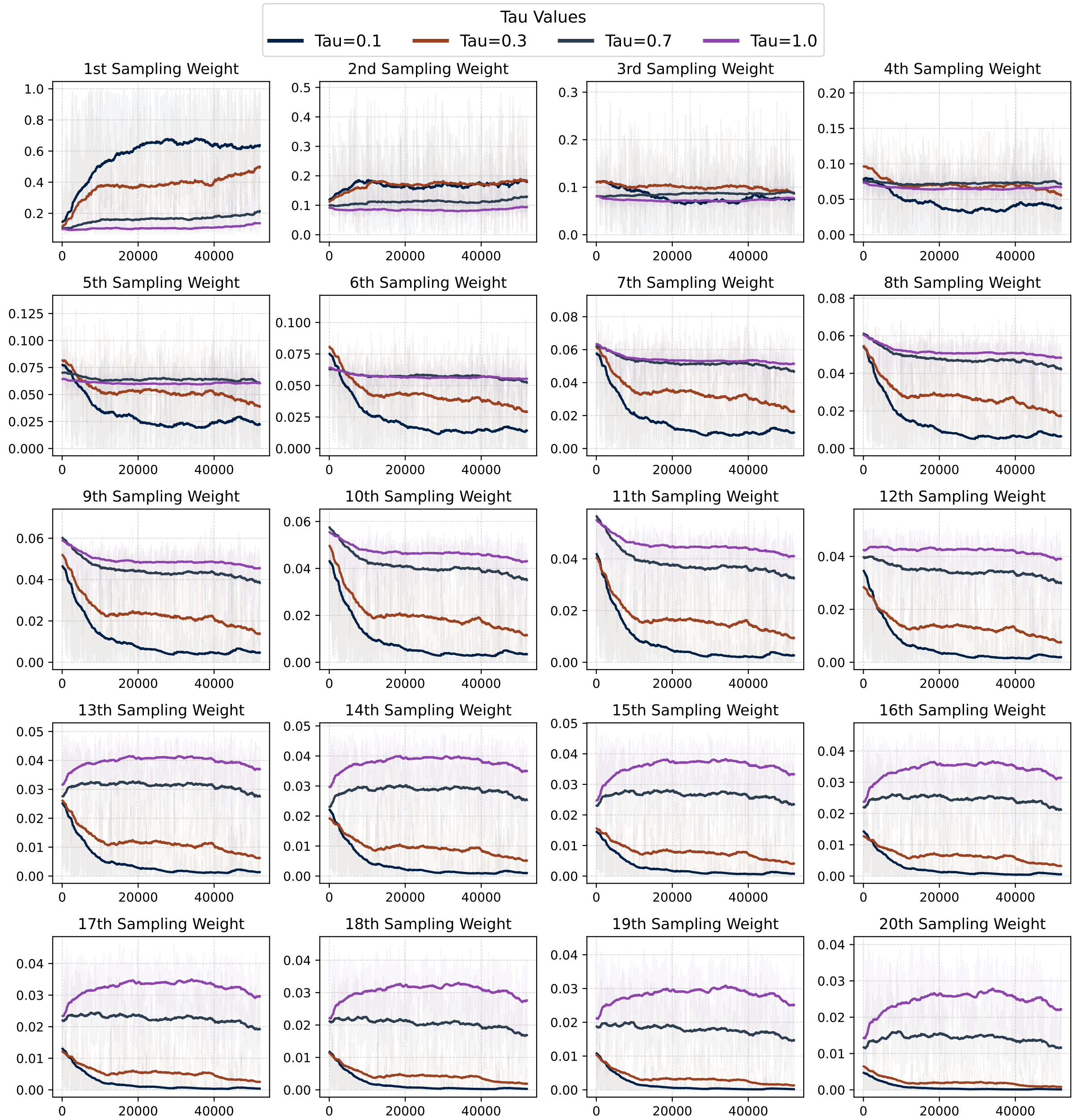

Figure 6: The impact of different τ values on the training process. We conduct with 20 candidate documents and RankT5 on the NQ dataset. The solid line in the figure represents the moving average. The differences in sampling weights indicate the Reranker's ability to distinguish between candidate documents.

the language model's perplexity when each document is used to condition the model's output. The KL divergence is minimized between the reranker's distribution over documents and the posterior distribution derived from the language model, which provides a direct measure of how much a document contributes to the model's performance. The target distribution for the reranker is computed as:

$$p_k = \frac{\exp(\log p_{\text{LM}}(\mathbf{a} \mid \mathbf{p}_k, \mathbf{q}))}{\sum_{i=1}^{K} \exp(\log p_{\text{LM}}(\mathbf{a} \mid \mathbf{p}_i, \mathbf{q}))}$$

The reranker is trained to minimize the KL-divergence between its predicted distribution over the documents and this target distribution:

$$\mathcal{L}_{\text{PDist}} = \sum_{k=1}^{K} p_{\mathcal{R}}(\mathbf{p}_k \mid \mathbf{q}) \log \frac{p_{\mathcal{R}}(\mathbf{p}_k \mid \mathbf{q})}{p_k}$$

Here, p_k is the distribution over documents that the language model would prefer, and the reranker is trained to match this distribution to improve the language model's perplexity. The KL-divergence loss encourages the reranker to select documents

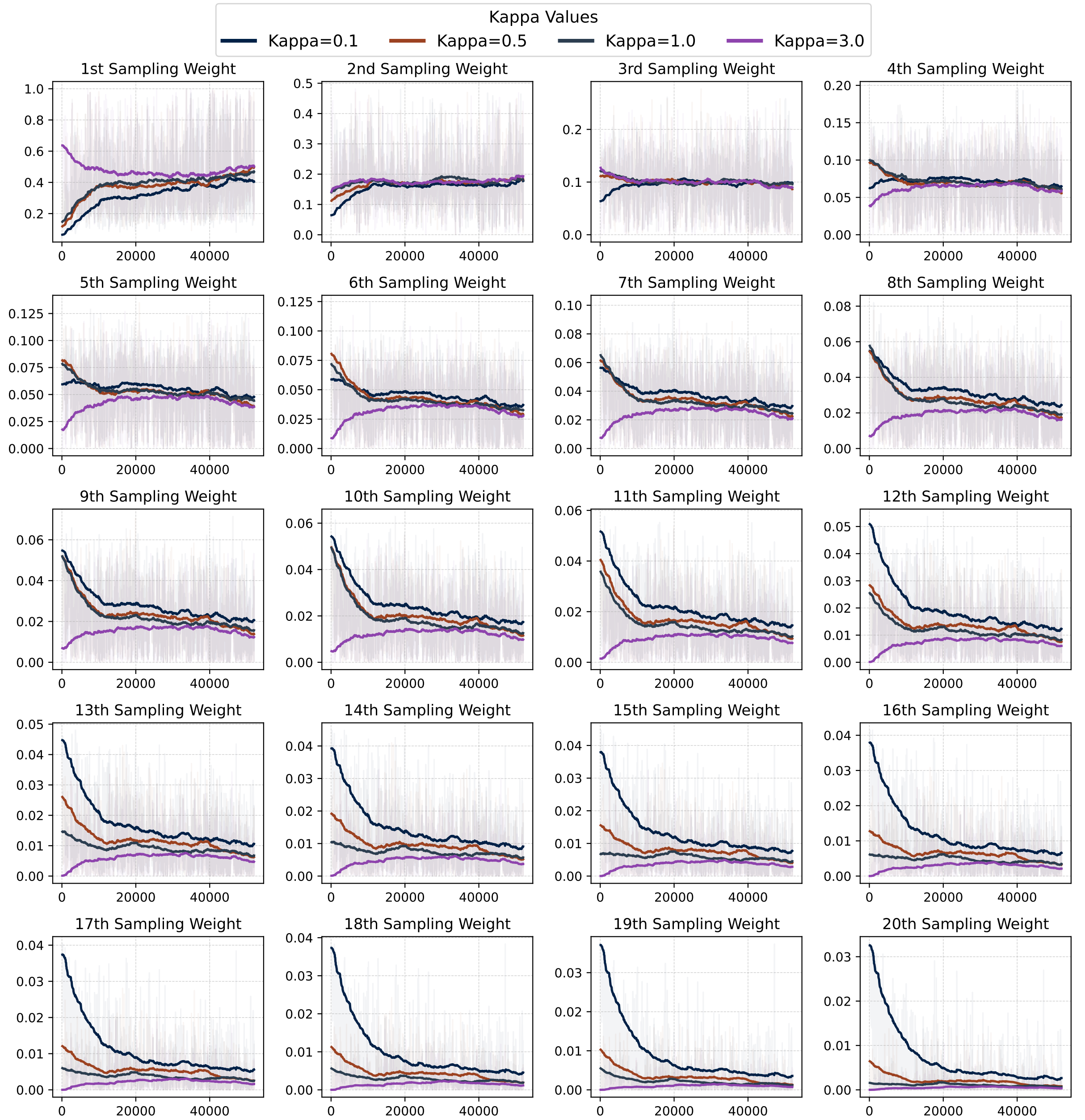

Figure 7: The impact of different κ values on the training process. We conduct with 20 candidate documents and RankT5 on the NQ dataset. The solid line in the figure represents the moving average. The differences in sampling weights indicate the Reranker's ability to distinguish between candidate documents.

that enhance the model's ability to generate the correct answer.

The objective function in EMDR[2] is based on maximizing the log-likelihood of the correct answer, given the query and documents. This method treats the documents as latent variables and aims to optimize the likelihood of generating the correct answer based on the combination of the language model and reranker's distributions. On the other hand, PDist focuses on optimizing the reranker's distribution by minimizing the KL-divergence between its predictions and the target distribution, which is derived from the language model's perplexity.

Leave-one-out Perplexity Distillation (LOOP) (Izacard et al., 2023): LOOP refines the PDist approach by considering the impact of each document in the context of all other documents in the top-k set. For each document, the log-likelihood of the output is computed by excluding the document from the retrieval set, and the negative of this value is used as a relevance score. The target distribution is:

$$p_{\text{LOOP}}(\mathbf{p}_k) = \frac{\exp(-\log p_{\text{LM}}(\mathbf{a} \mid \mathcal{D}_K \setminus \{\mathbf{p}_k\}, \mathbf{q}))}{\sum_{i=1}^{K} \exp(-\log p_{\text{LM}}(\mathbf{a} \mid \mathcal{D}_K \setminus \{\mathbf{p}_i\}, \mathbf{q}))}$$

The reranker is trained to minimize the KL-divergence between this distribution and the one obtained from the reranker.

C More Details

C.1 Variants of FiD

Recent advancements in Open-Domain Question Answering have led to the development of several enhanced Fusion-in-Decoder models. KG-FiD (Yu et al., 2022) enhances the traditional FiD framework by integrating knowledge graphs to establish structural relationships among retrieved passages. This integration employs graph neural networks to re-rank passages, selecting the most pertinent ones for answer generation, thereby improving both effectiveness and efficiency. FiDO (de Jong et al., 2023) addresses memory bandwidth constraints inherent in the FiD architecture by reallocating computational resources. This optimization results in a significant increase in inference speed without compromising performance, making it more suitable for real-time applications. FiD-Light (Hofstätter et al., 2023) focuses on efficient retrieval-augmented text generation by optimizing the balance between retrieval and generation components. This approach reduces computational overhead while maintaining answer accuracy, offering a more resource-efficient alternative. RFiD (Wang et al., 2023) introduces a multi-task learning approach to discern evidentiality, combining passage re-ranking with sentence classification. This method enhances the model's ability to identify causal relationships between questions and passages, leading to improved answer accuracy. Multi-Granularity Guided Fusion-in-Decoder (MG-FiD) (Choi et al., 2024) further refines the FiD approach by aggregating evidence across multiple levels of granularity. It harmonizes passage re-ranking with sentence-level classification, enhancing both accuracy and decoding efficiency.

In our experiments, since we are mainly focusing on reranker training strategies rather than reader, we utilize the classical Fusion-in-Decoder model architecture. Building upon this foundation, we compare our approach with various LLM-supervised reranker training strategies to assess their impact on ODQA performance.

C.2 Multi-Hop Question Answering

Multi-hop Question Answering (QA) systems typically follow a two-phase process: first, retrieving relevant passages, and then using these passages to answer the question. 2WikiHop (Ho et al., 2020), HotpotQA (Yang et al., 2018), Musique (Trivedi et al., 2022), and MultiHop-RAG (Tang and Yang, 2024) are widely used benchmarks for evaluating and improving RAG systems in handling complex multi-hop reasoning tasks. The retrieval strategies can differ depending on the QA setting, which may either be open-domain or reading comprehension.

In open-domain QA, the focus is on retrieving relevant passages from a large corpus. Methods like MDR (Xiong et al., 2021) and BeamDR (Zhao et al., 2021) are commonly used in this context. In the case of reading comprehension, retrieval methods are generally categorized into one-step and two-step approaches. One-step methods, such as SAE (Tu et al., 2020), rank passages by concatenating the question with each candidate passage. Two-step methods, including S2G (Wu et al., 2021) and FE2H (Li et al., 2023), start by selecting an initial hop passage and then refine the search by pairing it with additional candidates. The R^3 model (Yin et al., 2023) enhances this approach by selecting multiple passages at the outset and combining them to find the correct answer. Beam Retrieval (Zhang et al., 2024) further extends the process by using a beam search, enabling it to handle more complex multi-hop retrieval tasks that go beyond just two hops.

Our work focuses on a different scenario: we analyze the limitations of the LLM-Supervised Reranker Training strategy in widely used RAG systems for multi-hop question answering tasks (as detailed in Appendix D) and propose an end-to-end reranker training strategy based on Gumbel Subset Sampling, which is well-suited for multi-hop question answering tasks.

C.3 Dataset Description

The datasets in our study encompass a variety of challenges designed to assess different facets of question answering. **HotpotQA** (Yang et al., 2018) is a multi-hop QA dataset that requires reasoning over multiple Wikipedia articles to derive answers, emphasizing both factual retrieval and reasoning capabilities. Similarly, **2WikiHop** (Ho

	Mining Setting		Reranker Setting			Generator Setting		
	Recall@5	NDCG@5	Recall@5	NDCG@5	MRR	EM	SubEM	F1
Dataset: `Hotpotqa`								
Reranker: `Rank-T5`								
- EMDR (Lin et al., 2024)	78.8	81.1	79.5	81.1	95.8	58.3	64.6	73.1
- PDist (Glass et al., 2022)	70.9	73.9	71.4	73.6	92.6	57.8	64.0	72.5
- LOOP (Izacard et al., 2023)	72.2	75.4	73.4	75.7	93.7	58.0	64.2	72.7
- ADist (Izacard and Grave, 2021a)	72.2	74.1	72.5	73.8	90.5	56.5	62.6	71.1
- G-Rerank	**81.9**	**83.7**	**83.1**	**84.0**	**95.9**	**58.8**	**65.1**	**73.5**
Reranker: `BGE-Base`								
- EMDR (Lin et al., 2024)	78.4	81.1	78.8	80.7	95.9	58.6	64.8	73.4
- PDist (Glass et al., 2022)	76.7	79.8	78.5	80.7	**96.1**	58.7	64.9	73.4
- LOOP (Izacard et al., 2023)	76.0	79.0	77.1	79.2	95.3	58.3	64.5	73.0
- ADist (Izacard and Grave, 2021a)	78.5	80.7	79.3	80.9	95.1	58.2	64.4	73.0
- G-Rerank	**81.6**	**83.3**	**82.6**	**83.3**	95.8	**58.8**	**65.0**	**73.5**

Table 4: Experiments on HotpotQA using FiD-Base as reader. We consider the settings illustrated in Figure 3. The best performance is highlighted in bold, while the second-best performance is underlined.

et al., 2020) extends the complexity of multi-hop reasoning by introducing questions that necessitate navigating a knowledge graph, enhancing the evaluation of entity-based information retrieval. **Musique** (Trivedi et al., 2022) is designed to assess compositional reasoning by decomposing complex questions into a sequence of simpler sub-questions, providing a structured approach to multi-step reasoning evaluation. Meanwhile, **Natural Questions (NQ)** (Kwiatkowski et al., 2019) presents real-world search queries answered using long-form documents, challenging models to extract and summarize information from extensive contexts. Lastly, **TextbookQA (TQA)** (Kim et al., 2019) focuses on domain-specific comprehension, where questions require understanding of textbook-style knowledge, integrating both textual and diagrammatic content for a holistic assessment of contextual understanding and inferential capabilities.

C.4 Additional Results with FiD-Base

We present additional results on the HotpotQA dataset using FiD-Base as the reader, as shown in Table 4. Our method outperforms others across most metrics, demonstrating its efficacy.

D Challenges in Handling Indirectly Relevant Documents with EMDR/PDist

Both EMDR^2 (Sachan et al., 2021; Shi et al., 2024; Lin et al., 2024) and PDist (Izacard et al., 2023; Glass et al., 2022) are based on the premise of distilling the importance of individual documents in a multi-document retrieval and generation process. While effective for ranking directly relevant documents, both methods encounter challenges when dealing with *indirectly relevant documents*, which provide context but do not directly contain the answer.

In EMDR^2, the objective is to maximize the log-likelihood of generating the answer given the query and individual documents. The loss function is given by:

$$\mathcal{L}_{\text{EMDR}^2} = \log \left[\sum_{k=1}^{K} p_{\text{LM}}(\mathbf{a} \mid \mathbf{q}, \mathbf{p}_k) p_{\mathcal{R}}(\mathbf{p}_k \mid \mathbf{q}) \right]$$

where $p_{\text{LM}}(\mathbf{a} \mid \mathbf{q}, \mathbf{p}_k)$ represents the language model's probability of generating the answer conditioned on the query and document, and $p_{\mathcal{R}}(\mathbf{p}_k \mid \mathbf{q})$ is the reranker's preference for the k-th document. However, this approach assumes the independence of documents when generating the answer. Indirectly relevant documents, while crucial in providing context, do not appear to contribute meaningfully when evaluated independently. The importance of such documents can only be assessed when they interact with other documents in the evidence chain, making their relevance difficult to capture in this formulation.

Similarly, PDist minimizes the KL divergence between the reranker's distribution $p_{\mathcal{R}}(\mathbf{p}_k \mid \mathbf{q})$ and the distribution p_k derived from the language model's perplexity. p_k represents the distribution over documents that the language model prefers based on its perplexity improvement:

$$p_k = \frac{\exp(\log p_{\text{LM}}(\mathbf{a} \mid \mathbf{p}_k, \mathbf{q}))}{\sum_{i=1}^{K} \exp(\log p_{\text{LM}}(\mathbf{a} \mid \mathbf{p}_i, \mathbf{q}))}$$

In both methods, the language model $p_{\text{LM}}(\mathbf{a} \mid \mathbf{q}, \mathbf{p}_k)$ computes the probability of generating the answer based on the query and a single document $\mathbf{p}_k$, treating the document in isolation. This formulation assumes that each document, independently, provides enough information to determine the relevance to the query and the answer. However, in the case of indirectly relevant documents, this assumption breaks down. Indirectly relevant documents do not contain the answer directly but instead contribute to the reasoning process by supporting or contextualizing other documents. When evaluated alone, these documents may appear less relevant or even irrelevant, which undermines the effectiveness of both methods.

E Irrelevant Document Setting

Multi-hop question answering in `HotpotQA` involves synthesizing information from multiple documents to resolve a single query. Although these questions can be categorized into four major reasoning types—such as bridging intermediate information, comparing entities, verifying multiple properties, or inferring properties through a bridge—they all share the common requirement of gathering evidence across several sources. In this setting, identifying and assessing *indirectly relevant documents* can be instrumental for measuring how effectively a model captures the full chain of reasoning. Our approach defines an *indirectly relevant document* as any document labeled as relevant in the dataset yet not explicitly containing the final answer. This rule is rational in that it highlights the documents that contribute background or bridging information. However, this simple rule sometimes blurs the distinction between direct and indirect relevance. For instance, when a document only partially contains the answer, or when multiple sources each provide different fragments of a single reasoning chain (especially in question types like comparing entities or verifying multiple properties), all supporting documents will be classified as indirectly relevant by this rule. The concept of "partial" evidence is inherently difficult to categorize as either direct or indirect, and our rule consequently treats such "partial" evidence as indirectly relevant documents.

After processing the dataset, we observe that in the training set, the ratio of total queries to data entries that do contain *indirectly relevant documents* is 90,447 to 664,247. In the development set, this ratio is 7,405 to 5,966. The relatively large number of data entries that do contain indirectly relevant documents allows for a robust evaluation of a model's ability to retrieve and utilize such supporting evidence. Thus, this formulation not only aligns well with the structure of `HotpotQA` but also provides a meaningful benchmark for analyzing the effectiveness of different methods in capturing multi-hop dependencies beyond direct answer retrieval.

F Learnable Sampling Weights Setting

F.1 Scalar Relevance Baselines

In our thesis, we employ different scalar metrics to quantify the relevance of each candidate document in the RAG system. These metrics correspond to the different LLM-supervised reranker training methods, and they serve as proxies for the document's contribution to generating the correct answer. In particular, we consider the following three metrics:

Lowest Perplexity For methods such as EMDR^2 and Perplexity Distillation (PDist), each candidate document $\mathbf{p}_k$ is evaluated by combining it with the query $\mathbf{q}$ and computing the language model's negative log-likelihood of generating the ground-truth answer $\mathbf{a}$. Formally, the scalar relevance score is defined as:

$$s_{\text{perplexity}}(\mathbf{p}_k) = -\log p_{\text{LM}}(\mathbf{a} \mid \mathbf{q}, \mathbf{p}_k).$$

In this setting, a lower perplexity (i.e., a higher value of $s_{\text{perplexity}}$) indicates that the document better facilitates the generation of the correct answer, and is therefore considered more relevant.

Highest Attention Score The Attention Distillation (ADist) method evaluates relevance by feeding all candidate documents to the language model simultaneously and aggregating the cross-attention scores. For each document $\mathbf{p}_k$, the relevance score is computed by weighting the attention score α_k with the L2 norm of its corresponding value vector $\mathbf{v}_k$:

$$s_{\text{attn}}(\mathbf{p}_k) = \alpha_k \, \|\mathbf{v}_k\|_2.$$

Here, a higher attention score signifies that the language model assigns more importance to the document during answer generation, thereby indicating greater relevance.

Highest Leave-one-out Perplexity The Leave-one-out Perplexity Distillation (LOOP) method assesses the impact of each candidate document by measuring the degradation in the language model's performance when that document is excluded from

Algorithm 2 Learnable Sampling Weights Setting

```
procedure StochasticSubsetMask(document weights w_1, ..., w_n, temperature τ, scale factor κ, subset size k)
    for j = 1 to k do
        w̃_i = −log(−log(u_i)) + κ · w_i,    u_i ∼ U(0, 1)    ∀i ∈ [n]
        M̂^j = max(M̂^{j−1}, softmax((w̃_1, ..., w̃_n) / τ))    # M̂^0 = [0, ..., 0]
    end for
    return M̂^k                                          ▷ Return Relaxed top-k Mask
end procedure

Given a query q, answer a, and n retrieved passages p_1, ..., p_n
Initialization: Initialize learnable document weights w_1 = 0, w_2 = 0, ..., w_n = 0
for each training step do
    M̂ = StochasticSubsetMask(w_1, ..., w_n, τ, κ, k)
    Apply DMA(M̂) to obtain logits and language loss L_LM       ▷ subsection 3.3
    Update document weights w_1, ..., w_n with ∇_{w_1,...,w_n} L_LM   ▷ Gradient-based update
end for
```

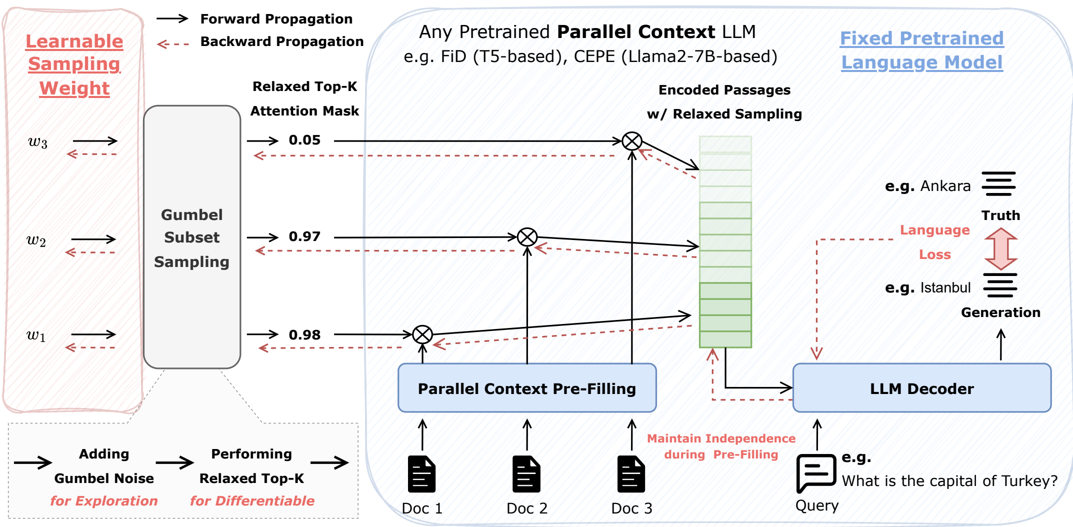

Figure 8: Setting of Learnable Sampling Weight. Optimizing candidate document sampling weights directly without leveraging reranker's prior textual knowledge.

the candidate set. For each document $\mathbf{p}_k$, the relevance score is defined as:

$$s_{\text{loop}}(\mathbf{p}_k) = -\log p_{\text{LM}}\Big(\mathbf{a} \mid \mathcal{D}_K \setminus \{\mathbf{p}_k\}, \mathbf{q}\Big),$$

where $\mathcal{D}_K$ denotes the set of top-k candidate documents. A higher leave-one-out score implies that the removal of $\mathbf{p}_k$ leads to a significant deterioration in the language model's ability to generate the answer, marking it as highly relevant.

F.2 Our method

In our proposed approach, we eliminate the dependency on a dedicated reranker by replacing it with a set of learnable scalar weights—one per document—that are initialized at zero and updated directly via gradients from the language modeling loss computed by the differentiable masked attention module. As detailed in Algorithm 2 and Figure 8, the algorithm employs stochastic Gumbel noise to perform a relaxed top-k selection over these weights, ensuring that the entire process remains fully differentiable. This method iteratively refines the document weights over multiple steps on a single query–answer pair and its corresponding documents, thereby enabling the model to learn which documents are most informative for the downstream language modeling task.

Dataset	URL	License
Multi-hop QA		
2WikiMultiHopQA (Ho et al., 2020)	https://github.com/Alab-NII/2wikimultihop	Apache License 2.0: https://github.com/Alab-NII/2wikimultihop/blob/master/LICENSE
HotpotQA (Yang et al., 2018)	https://hotpotqa.github.io/	CC BY-SA 4.0: https://hotpotqa.github.io/
MuSiQue (Trivedi et al., 2022)	https://github.com/stonybrooknlp/musique	CC BY 4.0: https://github.com/stonybrooknlp/musique/blob/main/LICENSE
Single-hop QA		
Natural Questions (NQ) (Kwiatkowski et al., 2019)	https://ai.google.com/research/NaturalQuestions	CC BY-SA 3.0: https://ai.google.com/research/NaturalQuestions/download
Textbook Question Answering (TQA) (Kim et al., 2019)	https://prior.allenai.org/projects/tqa	CC BY-NC 3.0: https://prior.allenai.org/projects/tqa

Table 5: Summary of URLs and Licenses for Datasets

G URLs and Licenses

Table 5 provides license information for the datasets we utilize in our experiments. We employ all the above datasets solely for research purposes, in accordance with their designated uses.