Tackling the Inherent Difficulty of Noise Filtering in RAG

Anonymous ACL submission

Abstract

Retrieval-Augmented Generation (RAG) has become a widely adopted approach to enhance Large Language Models (LLMs) by incorporating external knowledge and reducing hallucinations. However, noisy or irrelevant documents are often introduced during RAG, potentially degrading performance and even causing hallucinated outputs. While various methods have been proposed to filter out such noise, we argue that identifying irrelevant information from retrieved content is inherently difficult. This difficulty arises from the limitations of the Transformer architecture: the attention mechanism primarily models pairwise token interactions, which is insufficient for capturing complex, multi-token semantic relevance. Consequently, irrelevant documents are difficult to eliminate entirely, and LLMs must be robust against such noise. Moreover, we demonstrate that standard fine-tuning approaches are often ineffective in enabling the model to selectively utilize relevant information while ignoring irrelevant content due to the structural constraints of attention patterns. To address this, we propose a novel fine-tuning method designed to enhance the model's ability to distinguish between relevant and irrelevant information within retrieved documents. Extensive experiments across multiple benchmarks show that our approach significantly improves the robustness and performance of LLMs in noisy retrieval settings.

1 Introduction

Large Language Models (LLMs) (Brown, 2020) have demonstrated remarkable capabilities across a variety of tasks, including text generation and question answering (Ouyang et al., 2022; Wei et al., 2022), code generation (Gu, 2023), and information retrieval (Dai et al., 2024). However, current LLMs often suffer from serious hallucinations (Huang et al., 2023; Choudhary et al., 2025) due to a lack of factual information. Moreover, the knowledge embedded within LLMs is encoded in their parameters (Yang et al., 2024), meaning that incorporating new knowledge requires further fine-tuning, which is both time-consuming and resource-intensive. Consequently, augmenting LLMs with external retrievers has led to significant performance improvements (Lewis et al., 2020; Liang et al., 2024; Zhao et al., 2024; Izacard et al., 2023; Zhang et al., 2025b).

However, in real-world RAG scenarios, the information retrieved from documents is not always directly usable and often requires further processing because documents may contain noise information (Jiang et al., 2023b,a), and some documents may even be completely distracting, containing incorrect answers (Shi et al., 2023a; Wu et al., 2024; Zhang et al., 2025b; Ding et al., 2025). Such noise and distracting documents can negatively impact performance and some works attempt to train another filtering model to filter out noise and distracting documents (Xu et al., 2024; Yoran et al., 2023; Yan et al., 2024), but they show unsatisfying performance. In this paper, we show that the filtering process is inherently a complex problem, and limited number of transformer layers fail to solve the problem which means that noise is inevitable.

Considering the query *'Alice is exhausted but Bob is still excited; how does Bob feel?'* Here, *'exhausted'* is noise and should be excluded from the inference. However, assessing relevance necessitates considering the token *'Bob'* in the query alongside *'Alice'* the subject of *'exhausted'*. Thus, evaluating relevance requires the information from three or even more tokens. Yet, the attention mechanism typically computes only pair-wise relationships, making it challenging to resolve this issue within a limited number of transformer layers (Sanford et al., 2024). We further reveal that due to the autoregressive nature of large language models, placing the query ahead of the documents could be helpful because some information about the query could be encoded in the embedding of documents,

and this coule help to distinguish between relevant and irrelevant tokens.

Therefore filtering out noise documents is inherently difficult which explains why current retrieval methods (Izacard et al., 2021; Robertson et al., 2009; Karpukhin et al., 2020) would always incorporate some noise information in the retrieval results. Also training a model to filter out noise information fails to effectively reduce noise information as shown in Yoran et al. (2023). Therefore, it is important to train the model to make it more robust to noisy information. (Yoran et al., 2023; Ding et al., 2025). However, we show that fine-tuning methods like LoRA also struggle to filter out noise without compromising reasoning capacity because filtering irrelevant tokens while taking advantage of relevant tokens proves challenging.

For instance, assuming the parameters trained for all relevant documents are $\boldsymbol{W}_q$ and $\boldsymbol{W}_k$, we can hardly find $\Delta\boldsymbol{W}_q$ and $\Delta\boldsymbol{W}_k$ satisfying,

$$\begin{aligned} attn_{i,j} &= \sigma\left(\boldsymbol{x}_i^T(\boldsymbol{W}_q + \Delta\boldsymbol{W}_q)^T(\boldsymbol{W}_k + \Delta\boldsymbol{W}_k)\boldsymbol{x}_j\right) \\ &= \begin{cases} \epsilon & \text{noise} \\ \sigma\left(\boldsymbol{x}_i^T\boldsymbol{W}_Q^T\boldsymbol{W}_k\boldsymbol{x}_j\right) & \text{else,} \end{cases} \end{aligned}$$

where σ calculates the attention value after the softmax, and ϵ is a small number near 0. We show that this process can hardly be achieved. Then we propose a new method of fine-tuning to help better filtering out noise information and extensive experiments show that it works well.

The main contributions of this paper are:

- We reveal that the inherent triple-wise nature of the noise filtering process may necessitate numerous transformer layers for effective document preprocessing, which means we can hardly filter out noise documents with limited number of transformer layers.
- We show that fine-tuning the model to filter out noise while effectively taking advantage of the relevant information is challenging
- We developed a new fine-tuning method to better distinguish irrelevant tokens and extensive experiments shows its effectiveness.

2 Related Work

RAG in LLM. Many recent works have explored better RAG strategies. Shi et al. (2023b) treat the model as a black box and design a retriever to enhance performance. However, researchers have identified that noise in the prompt can negatively impact performance (Shi et al., 2023a; Zeng et al., 2025; Ding et al., 2025), some researchers focus on eliminating this noise information and compress the noise documents(Jiang et al., 2023b; Liu et al., 2025; Zhang et al., 2025a; Liskavets et al., 2025). And some others try to finetune the large language model to make it more robust to noise information(Yoran et al., 2023; Zhang et al., 2024, 2025b).

Filtering the noise. Yoran et al. (2023) tries to finetune the LLM to better filter out distracting documents, while RAFT (Zhang et al., 2024) uses different proportion of distracting documents to make it more robust. Self-RAG(Asai et al., 2023), use special tokens to represent that should the LLM use document informatione. Xu et al. (2024) first highlighted the duality of RAG, encompassing both benefits and detriments. Also, various researchers propose new fine-tuning methods to make the llm robust to noise information (Wang et al., 2023; Ding et al., 2025; Zhang et al., 2025b). Also Ding et al. (2025) finds that some specifically designed fine-tuning methods helps less in modern models. But they fail to understand why noise filtering is difficult in current LLMs.

3 Noise in the Document

In real-world applications of RAG, Due to the limited capacity of the retriever, distractions frequently arise in the retrieved documents (Shi et al., 2023a; Wu et al., 2024). Moreover, even in relevant documents, the presence of noise information is common (Jiang et al., 2023b,a; Xu et al., 2024). So we cannot directly use the information from retrieved documents; further processing is necessary because those noise and distracting information can be mistakenly interpreted as relevant, significantly degrading the performance of RALM. In this section, we examine the extent to which distracting and noisy information impacts overall performance of RAG.

For a token $\boldsymbol{x}$, we assume that its embedding contains two distinct pieces of information: $\boldsymbol{w}$, which acts as the regressor to determine the relevance of the document to the query, and $\boldsymbol{s}$, representing other information useful for inference. Thus, we can express the relationship as $H(\boldsymbol{x}) = H(\boldsymbol{s}, \boldsymbol{w})$, where $H(\cdot)$ denotes entropy. Let $\boldsymbol{z}$ be the token embedding after passing through the transformer

layers, and let $\boldsymbol{Z} = [\boldsymbol{z}_1^T, \ldots, \boldsymbol{z}_n^T]$, where n is the sequence length. We use $\boldsymbol{Z}_r$ to denote the embeddings of the related tokens.

When generating new tokens, the token embedding is integrated into the inference process. Therefore, if more information about $\boldsymbol{w}$ is incorporated into $\boldsymbol{z}$ (resulting in a high $I(\boldsymbol{w};\boldsymbol{z})$, $I(\cdot;\cdot)$ represents mutual information), the model can effectively recognize irrelevant tokens when conducting inference. Conversely, if the information is limited, the model is likely to be significantly affected by noise and distracting information in the input.

Theorem 3.1 (Impact of Noise)**.** *Given a set of input tokens, with δ percent of related and $1-\delta$ percent of distractions. If we assume documents contribute the same, and LLM conducts the filtering before inference, then*

$$L(\delta) - \mathcal{L}(1) \leq c_1 \sqrt{\left(\frac{(1-\delta)H(\boldsymbol{w}|\boldsymbol{z})}{\delta(I(\boldsymbol{w};\boldsymbol{z})+1)}\right) \cdot H(\boldsymbol{Z}_r)} + c_2,$$

where $\mathcal{L}(\delta)$ stands for the loss with δ percent of relevant documents and $\mathcal{L}(1)$ means the loss with all information related to the query, no distraction is involved. $\boldsymbol{z}$ is the embedding of documents, $H(\cdot)$ and $I(\cdot;\cdot)$ represents the entropy and mutual information. c_1, c_2 are constants and the value is shown in the Appendix.

From Equation 1, we see that the generalization error decreases as δ increases. This indicates an increasing number of distracting documents will have negative affect on performance. The term $H(\boldsymbol{Z}_r)$ represents the amount of relevant token information utilized to infer the final result, suggesting that the greater the reliance of the LLM on documents, the more detrimental the impact of noise becomes. Furthermore, performance is primarily determined by $I(\boldsymbol{w};\boldsymbol{z})$, which quantifies the information in the embedding relevant for assessing the document's relevance.

Apparently, to improve the performance, we can either reduce the number of distracting documents by more advanced retriever (large δ) or finetune the model to extract more information about distinguishing the relevance of token. In the following, we will show that these two methods fail because

- Filtering out irrelevant information is inherently difficult, limited number of transformer layers fail to solve it.
- Fine-tuning the model can hardly distinguish irrelevant information while taking advantage of relevant ones.

4 Filtering Noise Documents

Clearly, noise in the documents adversely affects the performance of large language models (LLMs). And various of researchers are trying to develop new retrieval methods to reduce the number of noise information, and some others try to develop a filtering model to filter out the irrelevant information before input to LLM. In this section, we show that filtering out noise documents is inherently a complex problem, limited number of transformer layers can hardly solve the problem.

4.1 The Triple-Wise Problem

For an input sequence $\boldsymbol{X} = [\boldsymbol{x}_0^T, \boldsymbol{x}_1^T, \ldots, \boldsymbol{x}_{n-1}^T]$, the vector $\boldsymbol{r}$ indicates the relevance of each token. Specifically, for each token $\boldsymbol{x}_i$, a relevance score of $r_i = 0$ signifies that the token is irrelevant to the query. It's crucial to note that calculating r_i does not solely depend on the token embedding $\boldsymbol{x}_i$ and the query; rather, it may require the involvement of three or more tokens. For instance, in the query *"Alice is exhausted, but Bob is still very excited, showing no signs of fatigue. How does Bob feel?"*, the word *"exhausted"* acts as noise and should be excluded during inference. However, determining relevance necessitates considering *"Bob"* in the query alongside *"Alice"*, the subject of *"exhausted"*. Therefore, identifying the relevance of a token demands information from multiple tokens, yet self-attention computes relationships only between pairs, making it challenging to address this issue within a single transformer layer.

Consider a simple scenario in which three tokens are sufficient to determine the relevance of a given token. The task is to ascertain whether there exists information indicating that token $\boldsymbol{x}_i$ is noise relative to the query. Let $g(\boldsymbol{x}_i, \boldsymbol{x}_a, \boldsymbol{x}_b)$ be the function designed to identify whether tokens $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$ suggest that token $\boldsymbol{x}_i$ is a relevant token, if $g(\boldsymbol{x}_i, \boldsymbol{x}_a, \boldsymbol{x}_b) = 0$, then x_i is a noise token, otherwise it is relevant. Thus, the problem can be framed as follows:

$$r_i = \begin{cases} 0 & \text{if } \exists\, a, b \; s.t. \; g(x_i, x_a, x_b) = 0, \\ 1 & \text{else.} \end{cases}$$

And we need a transformer based model $\mathcal{M}$ to calculate $\boldsymbol{r} = \mathcal{M}(\boldsymbol{X})$. Following (Sanford et al., 2024), we can show that one layer of transformer can hardly complete the task.

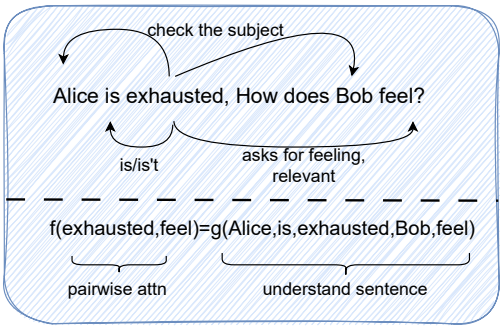

Figure 1: Judging the relevance of the token 'exhausted' actually requires to check the whole meaning of the sentence, so it can hardly be done by limited number of transformer layers.

Theorem 4.1. *For input documents of length n, and a constant c if $mpH \leq c \cdot nH(\boldsymbol{w})/\log\log n$, then there is no one layer transformer $\mathcal{M}$ with embedding size m, precision p and H heads to judge the relevance of tokens.*

Consequently, a single layer of multi-head attention struggles to assess the relevance of a token. As noted in Conjecture 19 of Sanford et al. (2024), even multiple layers of multi-head attention may not effectively resolve this issue. The challenge arises from the triple-wise nature of the problem contrasted with the pair-wise nature of attention; the model can only evaluate a token's relevance when its embedding contains substantial information. For example, as shown in Figure 1, to assess the token *"exhausted"* in the sentence *"Alice is exhausted. How does Bob feel?"*, the embedding must encompass information about its subject, *"Alice/Bob"*, as well as contextual details from phrases like *"is/isn't"* and *"feeling"*. Therefore, a significant amount of information must be incorporated into the embedding before any judgment can be made, it actually requires the understanding about the meaning of the whole sentence between judging the relevance of the token.

However, a single layer of self-attention can only consider the input $\sum_j a_{i,j}\boldsymbol{x}_j$, which contains information up to mp, where m represents the embedding dimension and p indicates precision. The term mp signifies the maximal entropy of the embedding, which means the maximal information the embedding can carry. But this can hardly represent the meaning of the whole sentence, especially when the input sequence is long.

This indicate that current retrieval methods like contriever (Izacard et al., 2021) would like to incorporate noise information in the retrieved documents. Golden retrieval is difficult unless a large model is used which is resource and time consuming.

4.2 The Auto-Regressive LLM

Judging the relevance of token is inherently a difficult problem, and here we show that it is even more difficult for current large language models. Currently, most LLMs are trained in an auto-regressive manner, meaning that a token cannot aggregate information from any tokens that appear later in the sequence; thus, $a_{i,j} = 0$ if $j > i$. Typically, the query is positioned after the document tokens, preventing these document tokens from assessing their relevance effectively because they have no access to the query. Consequently, the relevance judgment must occur during the calculation of the query embeddings, which is usually much shorter than the document, making the nose filtering harder because we need to judge the relevance of n_{doc} tokens during the calculation of n_{query} tokens.

Instead, if we position the query ahead of documents, then the relevance can be effectively calculated during the calculation document token embedding. This arrangement enables the information of query to be effectively transferred to the document tokens for relevance judgment. In this way we can judge the relevance of n_{doc} tokens during the calculation of n_{doc} tokens, as n_{doc} is usually mucn larger than n_{query}, so placing the query ahead could help. Therefore, we can hypothesize that when there is noise in the document, placing the query at the beginning would help the judgment.

We conduct experiments on various of models with NQ, TriviaQA, HotpotQA, 2Wikimultihopqa, and ASQA, more details about the experiment is shown in the Appendix. As shown in Table 1, positioning the query ahead significantly enhances performance for those models.

5 Fine-tuning Helps But not Perfectly

Clearly, noise in the documents negatively impacts the performance of LLMs, and noise information is inevitable in RAG. Therefore some researchers focus on fine-tuning the model to make it more robust when faced with noise information thereby enhancing performance (Yoran et al., 2023; Zhang et al., 2024; Jiang et al., 2023b). And there is a question that, since filtering out noise is inherently a difficult problem, can the finetuned LLM effec-

	NQ		TriviaQA		HotpotQA		2wiki		ASQA	
	reverse	vanilla	reverse	vanilla	reverse	vanilla	reverse	vanilla	reverse	vanilla
GPT-4o	**62.4**	61.1	**72.9**	67.2	**82.1**	81.7	**90.8**	88.4	**49.6**	47.5
Gemini 1.5 Pro	62.6	**63.1**	**69.2**	66.3	80.3	**81.0**	**85.5**	78.2	**46.9**	46.1
Claude3.7 Sonnet	**60.3**	59.0	**70.3**	66.7	**78.0**	77.8	**84.0**	81.0	**51.1**	50.6
Qwen2.5 72B	**58.0**	57.2	**66.0**	63.4	**79.6**	78.6	**84.6**	80.4	**48.5**	47.5
LLama3 70B	**58.8**	57.8	**68.8**	62.9	**79.7**	74.7	**81.6**	77.8	44.5	**46.2**
Qwen2.5 7B	**58.5**	54.3	**54.1**	54.1	71.0	**71.5**	**65.7**	63.8	**42.4**	41.3
Mistral 7B	**59.8**	59.2	64.0	**64.9**	**77.4**	77.0	**78.3**	72.8	**46.1**	45.8
llama3.1 8b	**52.1**	52.1	**58.5**	53.9	**71.0**	70.3	**63.3**	59.5	**44.6**	43.0

Table 1: The performance of 3000 samples except ASQA (948 samples), this shows that putting the query ahead could help the performance.

tively filter out irrelevant information and gather useful information to get the final answer?

Let $\boldsymbol{r}$ represents the relevance of tokens, $r_i = 0$ means that token $\boldsymbol{x}_i$ is a noise token, otherwise the token is relevant. Let $attn(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{W}_q\boldsymbol{x}_i)^T\boldsymbol{W}_k\boldsymbol{x}_j$ represent the attention pattern which is trained to extract relevant information. And we want to filter out irrelevant information as following:

$$\sigma\left(\widehat{attn}(\boldsymbol{x}_i, \boldsymbol{x}_j)\right) = \begin{cases} 0 & \text{if } r_j = 0, \\ \sigma\left((\boldsymbol{W}_q\boldsymbol{x}_i)^T\boldsymbol{W}_k\boldsymbol{x}_j\right) & \text{else,} \end{cases}$$

where σ means the softmax function. The desired attention pattern should effectively exclude noise while preserving the attention pattern of relevant tokens. Therefore, $\widehat{attn}$ can be considered the optimal response when confronted with noise, as it effectively filters out irrelevant tokens and utilizes the relevant information in the most efficient manner.

Fine-tuning the model involves adjusting its parameters and attention pattern, which allows the finetuned model to be expressed as

$$\begin{aligned} attn'(\boldsymbol{x}_i, \boldsymbol{x}_j) &= ((\boldsymbol{W}_q + \Delta\boldsymbol{W}_q)\boldsymbol{x}_i)^T\left(\boldsymbol{W}_k + \Delta\boldsymbol{W}_k\right)\boldsymbol{x}_j \\ &= \boldsymbol{x}_i^T(\boldsymbol{W} + \Delta\boldsymbol{W})\boldsymbol{x}_j, \end{aligned}$$

where $\boldsymbol{W} = \boldsymbol{W}_q^T\boldsymbol{W}_k$ and $\Delta\boldsymbol{W}$ represents the adjustments. The critical question arises: can we finetune the model to approximate the optimal one, i.e., is there a $\Delta\boldsymbol{W}$ such that $\sigma(attn'(\boldsymbol{x}_i, \boldsymbol{x}_j)) \approx \sigma(\widehat{attn}(\boldsymbol{x}_i, \boldsymbol{x}_j))$? $\sigma(\cdot)$ represents the softmax.

Theorem 5.1. *if there exists*

$$attn'(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i(\boldsymbol{W} + \Delta\boldsymbol{W})\boldsymbol{x}_j,$$

ϵ approximates $\widehat{attn}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ i.e.,

$$1 - \epsilon \leq \frac{\sigma(attn'(x_i, x_j))}{\sigma(\widehat{attn}(x_i, x_j))} \leq 1 + \epsilon,$$

where σ represents softmax, then we need

$$\xi_r \lesssim \ln\frac{1}{1-\epsilon},$$

where $\xi_r = \max(\boldsymbol{x}_i^T\Delta\boldsymbol{W}\boldsymbol{x}_j) - \min(\boldsymbol{x}_i^T\Delta\boldsymbol{W}\boldsymbol{x}_k)$, and $x_{j,k}$ is relevant tokens.

This shows that, to effectively fine-tune the model for filtering out irrelevant information in the attention matrix, we need $\xi_r \approx 0$. This implies that for all tokens to be retained, $\boldsymbol{x}_i^T\Delta\boldsymbol{W}\boldsymbol{x}_j$ must remain nearly constant. An solution could make $||\Delta\boldsymbol{W}||$ small or even 0, but this actually also requires to split the tokens to be retained and to be filtered, so a small $||\Delta\boldsymbol{W}||$ does not solve the problem as shown in the Appendix. As a result, approximating the optimal solution proves to be quite challenging because the attention will be disturbed due to the noise filtering fine-tuning.

Since the self-attention layer struggles to filter out irrelevant information while maintaining optimal reasoning performance, a pertinent question arises: **can the feed-forward network connected to the self-attention layer effectively filter out mistakenly incorporated information?** Here, we denote the information necessary to assess the relevance of the involved tokens as $\hat{\boldsymbol{w}}$, so $H(\hat{\boldsymbol{w}}) = H(\boldsymbol{w}_{i1}, \boldsymbol{w}_{i2}, \ldots)$, where $\boldsymbol{w}_{ij}$ represents the relevance information of the token aggregated by self-attention and those tokens form the input to the feed forward layer.

Theorem 5.2. *For a Feed Forward Network f and the input embedding v contains $1-\delta$ percent of noise information, assume the optimal function is*

$\hat{f}(x)$ which filter out the noise and finish the inference let $p = \Pr\left(||f(x) - \hat{f}(x)|| > t'\right)$,

$$t' = t + c_1\sqrt{(1-\delta)\frac{H(\hat{\boldsymbol{w}}|\boldsymbol{v}) - 1}{H(\hat{\boldsymbol{w}})} \cdot I(\boldsymbol{s};\boldsymbol{v})} + c_2,$$

where t is a constant, then

$$p > \frac{1}{C}\left(H(\boldsymbol{v}) - \delta\frac{I(\hat{\boldsymbol{w}};\boldsymbol{v}) + 1}{H(\hat{\boldsymbol{w}})} \cdot I(\boldsymbol{s};\boldsymbol{v})\right),$$

even if the model performs best when conduct filtering. $H(\boldsymbol{v})$ stands for all the information needed to conduct inference. $C = \log\frac{|\mathcal{V}|}{N_{max}(t)}$ is a constant and we show it in the Appendix.

From the above theorem, it is evident that achieving better performance requires the input to incorporate information that effectively distinguishes irrelevant tokens (a large $I(\hat{\boldsymbol{w}};\boldsymbol{v})$) and the necessary information for inference (a large $I(\boldsymbol{s};\boldsymbol{v})$). Additionally, there should be minimal noise incorporated into the input (a large δ).

As previously noted, the self-attention module struggles to filter out irrelevant information while retaining relevant tokens, which makes it difficult to increase δ. Furthermore, it is difficult to integrate additional information about $\hat{\boldsymbol{w}}$ (i.e., to increase $I(\hat{\boldsymbol{w}};\boldsymbol{v})$) because the input to the MLP layer is given by $\boldsymbol{v} = \sum_j a_{i,j}\boldsymbol{x}_j\boldsymbol{W}_v$, resulting in $\boldsymbol{v}$ being an embedding of dimension m and precision p. This limited dimensionality makes it challenging to encapsulate all the necessary information for filtering out irrelevant documents for all involved tokens. Consequently, the Feed Forward Layer is unlikely to achieve optimal performance when dealing with noisy input.

5.1 Fine-tuning for Noise Filtering

Conventional fine-tuning paradigms, primarily adjust the model's behavior by introducing an update matrix, $\Delta\boldsymbol{W}$, to the original weight matrix $\boldsymbol{W}$. The modified attention score between a query token i and a key token j is computed based on their embeddings, $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. However, as noted in Theorem 5.1, a simple linear addition of the update term $\boldsymbol{x}_i^T\Delta\boldsymbol{W}\boldsymbol{x}_j$ faces a fundamental trade-off: it struggles to simultaneously (1) create a sufficiently large margin between the attention scores of relevant and irrelevant tokens, and (2) preserve the nuanced, relative attention patterns among multiple relevant tokens, which is crucial for the model's intrinsic reasoning capabilities.

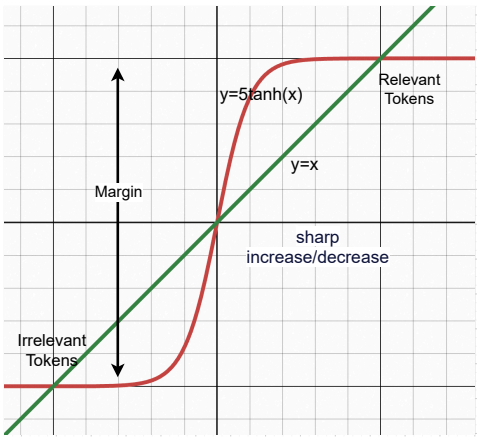

Figure 2: plot of $5\tanh(x)$, this shows that, by tanh, we can effectively enlarge the margin between relevant and irrelevant tokens and maintain similar attention weight to those relevant ones.

To address this, we propose to augment the standard attention mechanism with a non-linear rectification function, $g()$, applied to the attention update. The final attention is computed as:

$$attn'(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^T\boldsymbol{W}\boldsymbol{x}_j + g(\boldsymbol{x}_i^T\Delta\boldsymbol{W}\boldsymbol{x}_j). \quad (1)$$

The core of our method lies in the design of the rectification function g(x). The function g(x) is designed to operate in two distinct regimes: a filtering regime for low-relevance tokens and a refinement regime for high-relevance tokens. This is achieved by the following formulation:

$$g(x) = \begin{cases} \max(\xi \cdot \tanh(x), x) & \text{if } x >= 0, \\ \min(\xi \cdot \tanh(x), x) & \text{else,} \end{cases}$$

where ξ is a hyperparameter that controls the saturation threshold. The behavior of $g(x)$ is illustrated in Figure 2. For small or negative values of the attention update x, the term $\xi * tanh(x)$ dominates. The hyperbolic tangent function offers two key advantages:

- Sharp Discrimination: The steep gradient of $tanh(x)$ around $x = 0$ creates a sharp transition, effectively amplifying the margin between positive (potentially relevant) and negative (irrelevant) attention updates. This serves as a powerful mechanism for filtering out noise.

- Saturating Behavior: As x becomes sufficiently large, $tanh(x)$ approaches 1, causing

the output to saturate at ξ. This ensures that all highly relevant tokens receive a consistent and significant attention boost, preventing their original relative attention scores (after Softmax) from being severely distorted. The scaling factor ξ is chosen to be large enough to establish a clear margin, determined by the typical attention score variance in the base model.

In this way, we can effectively separate the relevant and irrelevant tokens during the sharp discrimination, and we can ensure that the relevant tokens share similar attention score.

However, simply using the $\xi \cdot tanh(x)$ may not be enough, a key limitation of using $\xi \cdot tanh(x)$ alone is that it clamps the attention boosts for all highly relevant tokens to a single value ξ, precluding any further fine-grained adjustments among them. Our composite function $g(x)$ overcomes this. When the attention update x is large enough to exceed $\xi \cdot tanh(x)$, $g(x)$ reverts to a linear identity which means $g(x) = x$. This linear growth phase allows the model to continue differentiating among highly relevant tokens, enabling it to learn a more optimal attention distribution for the specific downstream task, rather than merely preserving the original one.

From a regularization standpoint, our method mitigates overfitting to noisy signals in the training data. As discussed in Theorem 3.1, token embeddings can contain spurious correlations. An unconstrained linear update might learn to assign attention with high variance scores based on such noise. The saturating nature of the $tanh$ component in $g(x)$ acts as a "soft clamp" constraining these potentially extreme values within a bounded range (ξ). This restriction prevents the model from becoming overly sensitive to noisy features, thereby enhancing its robustness and generalization performance. For instance, an attention update that might vary between $\xi \pm \delta$ due to noise would be regularized to $\xi \pm \epsilon$, where $\epsilon << \delta$.

Also, $\max(\cdot)$ and $\min(\cdot)$ is not a continous function, so we use

$$g(x) = \begin{cases} \max(\xi \cdot \tanh(x), x) & \text{if } x >= 0, \\ \min(\xi \cdot \tanh(x), x) & \text{else,} \end{cases}$$
$$\approx \log(\exp(a) + \exp(b) + 1) - \log(\exp(-a), \exp(-b) + 1),$$

where $a = \xi \cdot tanh(x)$, $b = x$. Apparently, when $a, b >= 0$, $g(x)$ will be dominated by $\log(\exp(a) + \exp(b) + 1)$, which is approximately $max(a, b)$. Otherwise if $a, b < 0$, $g(x)$ will be dominated by $-\log(\exp(-a), \exp(-b) + 1)$, which is approximately $min(a, b)$.

6 Experiments

6.1 Datasets and Metrics

To evaluate the performance of our proposed method, we use nature question (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017) which is traditionally used to evaluate the noise robustness of RAG system. Also, we use multi hop reasoning datasets HotpotQA (Yang et al., 2018) and 2WikiMultiHopQA (Ho et al., 2020) as well as the long form QA dataset ASQA (Stelmakh et al., 2022) to show the performance of our method. For the first 4 datasets, we use accuracy to measure the performance which is determined by whether the predicted answer contains the ground-truth answer. For ASQA, we use measure the percentage of short answers are shown in the generated answer to evaluate the performance.

For all 5 datasets, we use BM25 as the retriever and retrieve top 3 noise documents together with the gold documents as the input to the LLM, showing that our method can effectively distinguish distracting documents.

For the first 4 datasets, we randomly select 3000 samples to test and another 7000 to train. For ASQA we use the split of ALCE (Gao et al., 2023) and use the 948 samples to test the performance and another 4000 for training.

More Experimental settings can be seen in the Appendix A

6.2 Implementation Details

When calculating $\Delta \boldsymbol{W}$, we use Low Rank Adaption, which means we actually calculates $\Delta \boldsymbol{W} = \boldsymbol{A} \cdot \boldsymbol{B}$, $\boldsymbol{A} \in \mathbb{R}^{h \times r}$, where h is the hidden dimension and r is the rank, in our experiments, we set $r = 64$.

We compare our method with vanilla LoRA finetuning, which means

$$attn_{i,j} = \left(\boldsymbol{x}_i^T (\boldsymbol{W}_q + \Delta \boldsymbol{W}_q)^T (\boldsymbol{W}_k + \Delta \boldsymbol{W}_k) \boldsymbol{x}_j\right).$$

And directly modeling $\Delta \boldsymbol{W}$ instead of ΔQ and ΔK, which means,

$$attn_{i,j} = \boldsymbol{x}_i^T \boldsymbol{W}_q^T \boldsymbol{W}_k \boldsymbol{x}_j + \boldsymbol{x}_i^T \Delta \boldsymbol{W}_q^T \Delta \boldsymbol{W}_k \boldsymbol{x}_j.$$

		NQ		TriviaQA		HotpotQA		2wiki		ASQA	
		reverse	vanilla	reverse	vanilla	reverse	vanilla	reverse	vanilla	reverse	vanilla
Llama8B	vanilla	52.1	52.1	58.5	53.9	71.0	70.3	63.3	59.5	44.6	43.0
	LoRA	66.3	65.0	73.3	72.7	84.6	83.8	95.6	94.5	48.8	48.0
	$\xi = 0$	66.1	64.1	73.6	72.7	84.6	83.8	96.0	94.5	49.8	47.8
	$\xi = 1$	68.3	66.3	73.8	73.4	84.6	83.5	96.9	96.4	50.5	48.2
	$\xi = 3$	**68.6**	67.1	74.3	**74.8**	84.9	84.1	**96.8**	**97.0**	51.5	**48.7**
	$\xi = 5$	68.5	**68.4**	73.5	74.4	**86.1**	83.1	96.8	96.0	**52.3**	48.3
	$\xi = 10$	66.9	65.1	**74.4**	74.0	84.4	**84.2**	96.4	96.4	51.2	48.7

Table 2: Performance of our fine-tuning method when faced with explicit distracting documents. It is worth noting that $\xi = 0$ results in $g(x) = x$

And our method, which adds the tanh activation function

$$attn_{i,j} = \boldsymbol{x}_i^T \boldsymbol{W}_q^T \boldsymbol{W}_k \boldsymbol{x}_j + g\left(\boldsymbol{x}_i^T \Delta \boldsymbol{W}_q^T \Delta \boldsymbol{W}_k \boldsymbol{x}_j\right).$$

When calculating $\Delta \boldsymbol{W}_k$, we use Group Query Attention like used in the LLaMA architecture. And those $\Delta \boldsymbol{W}$ are approximated by Low Rank Adaption with rank 64, so the parameters for training are the same. And we conduct our experiments on Llama-3.1-8B-Instruct.

6.3 Main Results

Table 2 shows the performance when faced with explicit distracting documents, we evaluate two different setting where the query is placed behind the documents or ahead the documents. The result shows that with $g(x)$, the model can better distinguish between relevant and distracting documents, showing better performance. Also, we can observe that after fine-tuning, placing the query ahead of documents still helps, which further validate our theory in Section 4.2. Also, we can find that directly modeling $\Delta \boldsymbol{W}$ performs similarly with LoRA fine-tuning, this is not surprising because they are technically equal, and they use the same number of parameters.

And we can observe that the hyperparameter ξ actually does not require specific tuning, we can directly set the value based on the inherent attention weight margin of the model. We show the margin of the original attention weight of Llama 3.1 8B in Figure 3, we can observe that most of the margins of Llama lies about 6, so with $\xi = 3$, (margin between (-3,3)), the model can effectively distinguish irrelevant tokens, and larger values like $\xi = 5$ also performs good.

We also conduct experiments when $g(x) = \xi \cdot tanh(x)$ and $g(x) = \xi \cdot tanh(x) + x$. $g(x) = tanh(x)$ stands for the situation that $g(x)$ only focus on enlarge the margin between relevant and irrelevant tokens without further linear growth. And $g(x) = tanh(x) + x$ stands for the situation where the the steady growth process is missing, so the soft clamp will not work. We also try using $g(x) = max(x, \xi \cdot (sigmoid(x) - 0.5))$ to show the performance when using sigmoid instead of tanh. As shown in Figure 4, simply use $tanh(x)$ or $tanh(x) + x$ has suboptimal performance and replacing $tanh$ with $sigmoid(x) - 0.5$ also helps the performance. We also discuss the efficiency of our method in Appendix A

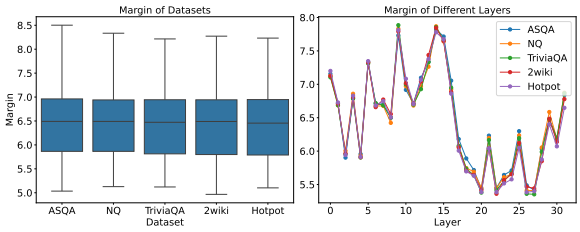

Figure 3: The margin of attention scores. We calculate the margin as the difference between the 90th and 10th percentile attention scores to reduce the influence of outliers

7 Conclusion

In this paper, we highlight that noise filtering in RAG is inherently a difficult problem, judging is a token relevant may involve three or even more other tokens, and limited number of transformer layers can not effectively solve it. Then we show that simply fine-tuning the LLM may not be optimal as it will disturb the attention pattern, which could lead to worse performance. Then we propose a new fine-tuning method which can help to better filter out noise and take advantage of the relevant information, extensive experiments show that our method works well.

Limitations

The paper discusses the limitation of LLM when dealing with noise information, showing that current LLMs can not effectively process noise information. However, although a new fine-tuning method is proposed, it can not solve the triple-wise problem, completely solving the noise in RAG system may require more advanced architecture which requires further exploration.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*.

Tom B Brown. 2020. Language models are few-shot learners. *arXiv preprint ArXiv:2005.14165*.

Sarthak Choudhary, Nils Palumbo, Ashish Hooda, Krishnamurthy Dj Dvijotham, and Somesh Jha. 2025. Through the stealth lens: Rethinking attacks and defenses in rag. *arXiv preprint arXiv:2506.04390*.

Sunhao Dai, Chen Xu, Shicheng Xu, Liang Pang, Zhenhua Dong, and Jun Xu. 2024. Bias and unfairness in information retrieval systems: New challenges in the llm era. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6437–6447.

Hanxing Ding, Shuchang Tao, Liang Pang, Zihao Wei, Liwei Chen, Kun Xu, Huawei Shen, and Xueqi Cheng. 2025. Revisiting robust rag: Do we still need complex robust training in the era of powerful llms? *arXiv preprint arXiv:2502.11400*.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023. Enabling large language models to generate text with citations. *arXiv preprint arXiv:2305.14627*.

Qiuhan Gu. 2023. Llm-based code generation method for golang compiler testing. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 2201–2203.

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*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and 1 others. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.

Gautier Izacard, Patrick 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. *Journal of Machine Learning Research*, 24(251):1–43.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023a. Llmlingua: Compressing prompts for accelerated inference of large language models. *arXiv preprint arXiv:2310.05736*.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023b. Longllmlingua: Accelerating and enhancing llms in long context scenarios via prompt compression. *arXiv preprint arXiv:2310.06839*.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

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

Kenji Kawaguchi, Zhun Deng, Xu Ji, and Jiaoyang Huang. 2023. How does information bottleneck help deep learning? In *International Conference on Machine Learning*, pages 16049–16096. PMLR.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, and 1 others. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Lei Liang, Mengshu Sun, Zhengke Gui, Zhongshu Zhu, Zhouyu Jiang, Ling Zhong, Yuan Qu, Peilong Zhao, Zhongpu Bo, Jin Yang, and 1 others. 2024. Kag: Boosting llms in professional domains via knowledge augmented generation. *arXiv preprint arXiv:2409.13731*.

Barys Liskavets, Maxim Ushakov, Shuvendu Roy, Mark Klibanov, Ali Etemad, and Shane K Luke. 2025. Prompt compression with context-aware sentence

encoding for fast and improved llm inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 24595–24604.

Jiaheng Liu, Dawei Zhu, Zhiqi Bai, Yancheng He, Huanxuan Liao, Haoran Que, Zekun Wang, Chenchen Zhang, Ge Zhang, Jiebin Zhang, and 1 others. 2025. A comprehensive survey on long context language modeling. *arXiv preprint arXiv:2503.17407*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Stephen Robertson, Hugo Zaragoza, and 1 others. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Clayton Sanford, Daniel J Hsu, and Matus Telgarsky. 2024. Representational strengths and limitations of transformers. *Advances in Neural Information Processing Systems*, 36.

Jonathan Scarlett and Volkan Cevher. 2019. An introductory guide to fano's inequality with applications in statistical estimation. *arXiv preprint arXiv:1901.00555*.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed Chi, Nathanael Schärli, and Denny Zhou. 2023a. Large language models can be easily distracted by irrelevant context. *Preprint*, arXiv:2302.00093.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023b. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. 2022. Asqa: Factoid questions meet long-form answers. *arXiv preprint arXiv:2204.06092*.

Zhiruo Wang, Jun Araki, Zhengbao Jiang, Md Rizwan Parvez, and Graham Neubig. 2023. Learning to filter context for retrieval-augmented generation. *arXiv preprint arXiv:2311.08377*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Siye Wu, Jian Xie, Jiangjie Chen, Tinghui Zhu, Kai Zhang, and Yanghua Xiao. 2024. How easily do irrelevant inputs skew the responses of large language models? *Preprint*, arXiv:2404.03302.

Shicheng Xu, Liang Pang, Huawei Shen, and Xueqi Cheng. 2024. Unveil the duality of retrieval-augmented generation: Theoretical analysis and practical solution. *arXiv preprint arXiv:2406.00944*.

Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. 2024. Corrective retrieval augmented generation. *arXiv preprint arXiv:2401.15884*.

Hongkang Yang, Zehao Lin, Wenjin Wang, Hao Wu, Zhiyu Li, Bo Tang, Wenqiang Wei, Jinbo Wang, Zeyun Tang, Shichao Song, and 1 others. 2024. Memory3: Language modeling with explicit memory. *arXiv preprint arXiv:2407.01178*.

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. *Preprint*, arXiv:1809.09600.

Andrew Chi-Chih Yao. 1979. Some complexity questions related to distributive computing(preliminary report). In *Proceedings of the Eleventh Annual ACM Symposium on Theory of Computing*, STOC '79, page 209–213, New York, NY, USA. Association for Computing Machinery.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2023. Making retrieval-augmented language models robust to irrelevant context. *arXiv preprint arXiv:2310.01558*.

Linda Zeng, Rithwik Gupta, Divij Motwani, Diji Yang, and Yi Zhang. 2025. Worse than zero-shot? a fact-checking dataset for evaluating the robustness of rag against misleading retrievals. *arXiv preprint arXiv:2502.16101*.

Gongbo Zhang, Zihan Xu, Qiao Jin, Fangyi Chen, Yilu Fang, Yi Liu, Justin F Rousseau, Ziyang Xu, Zhiyong Lu, Chunhua Weng, and 1 others. 2025a. Leveraging long context in retrieval augmented language models for medical question answering. *npj Digital Medicine*, 8(1):239.

Qianchi Zhang, Hainan Zhang, Liang Pang, Ziwei Wang, Hongwei Zheng, Yongxin Tong, and Zhiming Zheng. 2025b. Finefilter: A fine-grained noise filtering mechanism for retrieval-augmented large language models. *arXiv preprint arXiv:2502.11811*.

Tianjun Zhang, Shishir G Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E Gonzalez. 2024. Raft: Adapting language model to domain specific rag. *arXiv preprint arXiv:2403.10131*.

Penghao Zhao, Hailin Zhang, Qinhan Yu, Zhengren Wang, Yunteng Geng, Fangcheng Fu, Ling Yang, Wentao Zhang, and Bin Cui. 2024. Retrieval-augmented generation for ai-generated content: A survey. *arXiv preprint arXiv:2402.19473*.

A Experiments

A.1 Experiment settings

When conduct fine-tuning, we use learning rate of 1e-4, and we use kaiming initialization to initialize the parameter with $a = \sqrt{5}$. The experiments is conducted on 8 NVIDIA A100 80GB.

For NQ, TriviaQA, HotpotQA and 2Wiki, we randomly select 3000 samples to test and another 7000 to train. For ASQA we use the split of ALCE (Gao et al., 2023) and use the 948 samples to test the performance and another 4000 for training. Also, we train 3 epochs for each dataset except ASQA due to its limited number of data, we train it for 5 epochs instead. We train the model with batch size 8.

We use BM25 as the retriever and retrieve top 3 noise documents as noise. Then we mix it with the gold documents and shuffle the documents randomly as the input context.

A.2 Efficiency

However, although the new fine-tuning method could improve the performance, it requires to calculate two distinct attention matrix, while LoRA fine-tuning only requires to calculate one, which makes the method less efficient when conducting inference. We show the time consumption when conducting inference on the test set in Table 3, we can observe that our method cost longer when conducting inference, but the memory consumption is close because the extra attention matrix is added to the original one, so we do not need to keep 2 attention matrix. Also, the time consumption can be reduced, we can actually distill the performance into LoRA fine-tuning, specifically, when conduct LoRA fine-tuning, we add another MSE regularization term to make the extra attention weight of LoRA similar to the one with our fine-tuning method. The result is shown in Table 4.

		NQ	TriviaQA	HotpotQA	2Wiki	ASQA
Time	vanilla	13.84	10.8	12.24	15.84	34.8
	ours	19.2	15.84	19.36	22.4	44
Memory	vanilla	28.2	28.6	31.3	31.3	41.6
	ours	29.6	29.1	31.5	31.5	43.5

Table 3: The time and memory consumption (minutes, GB) during inference on test set. Our method costs more time, but we show in Table 4 that the time consumption can be reduced by distilling our method to vanilla LoRA.

	NQ	TriviaQA	Hotpot	2wiki	ASQA
LoRA	66.3	73.3	84.6	95.6	48.8
g(x)	68.6	74.3	84.9	96.8	51.5
LoRA+	68.5	74.2	84.9	96.9	51.3

Table 4: The performance of LoRA with regularization when $\xi = 3$. LoRA+ stands for the one with regularization.

B Proof of Section 3

B.1 Proof of Theorem 3.1

Assumption B.1. The model conducts filtering before inference, and those information regarded as noise would not be used during inference. When there are δ percent of relevant tokens and $1 - \delta$ percent of noise information are included in the inference, let $\boldsymbol{X}_r$ and $\boldsymbol{X}_n$ represents the relevant tokens and noise, $\boldsymbol{X}$ represents the input to the inference, then $I(\boldsymbol{X}_r; \boldsymbol{X}) = \delta H(\boldsymbol{X})$ and $I(\boldsymbol{X}_n; \boldsymbol{X}) = (1 - \delta)H(\boldsymbol{X})$

Assume that there are δ percent of tokens are actually related to the query, and other $1 - \delta$ percent tokens are distractions. In addition, with probability p_e the model mistakenly classify is the document

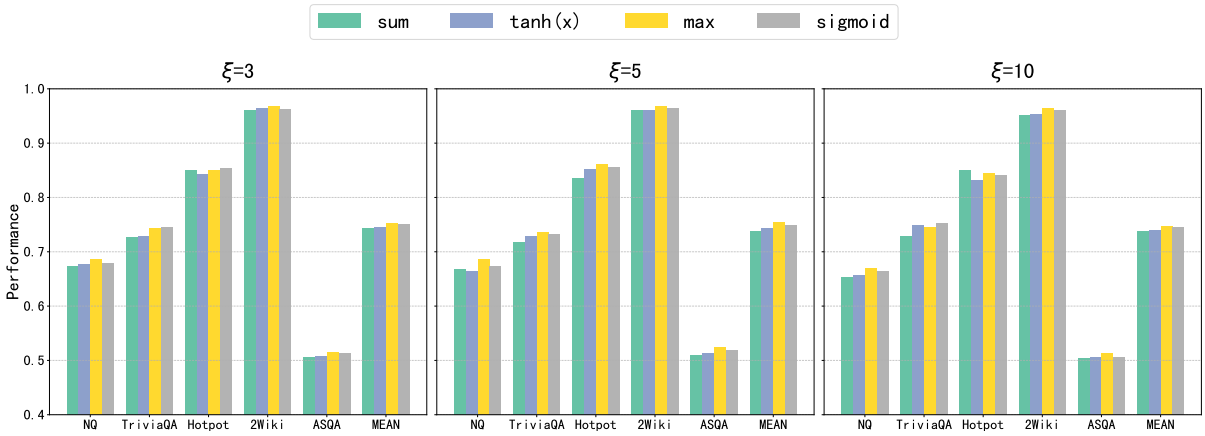

Figure 4: Compare between $max(\xi{\cdot}tanh(x), x)$, $\xi{\cdot}tanh(x)$ and $sum(\xi{\cdot}tanh(x), x)$. We also show the performance when we use $g(x) = max(x, 2\xi \cdot (sigmoid(x) - 0.5))$

relevant or not. Then, after filtering, $\delta \cdot (1 - p_e)$ percent of relevant tokens and $(1 - \delta) \cdot p_e$ distractions are left in the input. Then the percentage of distraction is,

$$\alpha = \frac{(1-\delta)\cdot p_e}{(1-\delta)\cdot p_e + \delta \cdot (1-p_e)} = \frac{(1-\delta)p_e}{p_e + \delta - 2\delta p_e} = \frac{1-\delta}{\frac{\delta}{p_e} - 2\delta + 1}$$

Based on Fano's inequality, $p_e \geq \frac{H(\boldsymbol{w}|\boldsymbol{z})-1}{H(\boldsymbol{w})}$, $\boldsymbol{z}$ is the embedding obtained by transformer, and we assume that $\boldsymbol{s}$ is the information we need to extract from document for further generation, $\boldsymbol{w}$ is the information to identify the relevance of tokens.

$$H(\boldsymbol{w}|\boldsymbol{z}) = H(\boldsymbol{w}) - I(\boldsymbol{w};\boldsymbol{z})$$
$$p_e \geq \frac{H(\boldsymbol{w}) - I(\boldsymbol{w};\boldsymbol{z}) - 1}{H(\boldsymbol{w})}$$
$$\alpha \geq \frac{1-\delta}{\frac{\delta H(\boldsymbol{w})}{H(\boldsymbol{w}) - I(\boldsymbol{w};\boldsymbol{z}) - 1} - 2\delta + 1}$$

Theorem B.2 (Theorem 1 of (Kawaguchi et al., 2023))**.** *Let $l \in \{1, \ldots, D\}$. Suppose that ϕ_l^s is fixed independently of the training dataset s. Then, for any $\delta > 0$, with probability at least $1 - \delta$, the following holds:*

$$\Delta(s) \leq G_3^l \sqrt{\frac{I(X; Z_l|Y)\ln(2) + G_2^l}{n}} + \frac{G_1^l(0)}{\sqrt{n}},$$

where $G_1^l(0) = \mathcal{O}(1)$, $G_2^l = \mathcal{O}(1)$, and $G_3^l = \mathcal{O}(1)$, as $n \to \infty$.

The theorem above regard the first l layers of a model as a encoder, and the rest as decoder. Consider the best Z_l where minimal information about $I(X; Z_l|Y)$ is kept, consider the best case where the noise is already filtered. In this way, Z_l is actually the output embeddings of tokens after filtering, and $I(X; Z|Y) = H(\boldsymbol{Z}_d) = \frac{\alpha}{1-\alpha} H(\boldsymbol{Z}_r)$, where $H(\boldsymbol{Z}_d)$ stands for the entropy of those distracting tokens and $H(\boldsymbol{Z}_r)$ stands for the entropy of those relevant tokens.

$$\begin{aligned}
\frac{\alpha}{1-\alpha} &= \frac{(1-\delta)p_e}{p_e + \delta - 2\delta p_e} \cdot \frac{p_e + \delta - 2\delta p_e}{p_e + \delta - 2\delta p_e - (1-\delta)p_e} \\
&= \frac{(1-\delta)p_e}{p_e + \delta - 2\delta p_e - (1-\delta)p_e} \\
&= \frac{(1-\delta)p_e}{\delta - \delta p_e} = \frac{1-\delta}{\frac{\delta}{p_e} - \delta}
\end{aligned}$$

Also,

$$
\begin{aligned}
I(X;Z|Y) &= \frac{\alpha}{1-\alpha} \cdot H(\boldsymbol{Z}_r) \\
&= \left(\frac{(1-\delta)(H(\boldsymbol{w}|\boldsymbol{z}) - 1)}{\delta(I(\boldsymbol{w};\boldsymbol{z}) + 1)} \right) \cdot H(\boldsymbol{Z}_r)
\end{aligned} \tag{2}
$$

Using Theorem B.2, we can observe that,

$$
\begin{aligned}
\Delta(s) &\leq G_3 \sqrt{\frac{I(X;Z|Y)\ln(2) + G_2}{n}} + \frac{G_1(0)}{\sqrt{n}} \\
&\leq G_3 \sqrt{\frac{I(X;Z|Y)\ln(2)}{n}} + \sqrt{\frac{G_2}{n}} + \frac{G_1(0)}{\sqrt{n}} \\
&= c_1 \sqrt{I(X;Z|Y)} + c_2 \\
&= c_1 \sqrt{\left(\frac{(1-\delta)(H(\boldsymbol{w}|\boldsymbol{z}) - 1)}{\delta(I(\boldsymbol{w};\boldsymbol{z}) + 1)} \right) \cdot H(\boldsymbol{Z}_r)} + c_2 \\
&\leq c_1 \sqrt{\left(\frac{(1-\delta)H(\boldsymbol{w}|\boldsymbol{z})}{\delta(I(\boldsymbol{w};\boldsymbol{z}) + 1)} \right) \cdot H(\boldsymbol{Z}_r)} + c_2
\end{aligned} \tag{3}
$$

where $c_1 = G_3\sqrt{\frac{\ln 2}{n}}$, $c_2 = \sqrt{\frac{G_2}{n}} + \frac{G_1(0)}{\sqrt{n}}$

$$
\mathcal{L}(\delta) - \mathcal{L}(1) \leq c_1 \sqrt{\left(\frac{(1-\delta)H(\boldsymbol{w}|\boldsymbol{z})}{\delta(I(\boldsymbol{w};\boldsymbol{z}) + 1)} \right) \cdot H(\boldsymbol{Z}_r)} + c_2, \tag{4}
$$

Theorem B.3 (Impact of Distractions)**.** *Given a set of documents, with δ percent of related and $1-\delta$ percent of distractions. If we assume documents contribute the same, and we ask the LLM to ignore the irrelevant documents, then,*

$$
\mathcal{L}(\delta) - \mathcal{L}(1) \leq c_1 \sqrt{\left(\frac{(1-\delta)H(\boldsymbol{w}|\boldsymbol{z})}{\delta(I(\boldsymbol{w};\boldsymbol{z}) + 1)} \right) \cdot H(\boldsymbol{Z}_r)} + c_2, \tag{5}
$$

where $\mathcal{L}(\delta)$ stands for the loss with $(1-\delta)$ distractions and $\mathcal{L}(1)$ means the training loss as in the training stage, no distraction is involved. $I(\boldsymbol{Z}_d;\boldsymbol{Y})$ is the mutual information between the embedding of related documents and the final embedding, this shows how much does LLM rely on documents during inference.

C Proof of Section 4

C.1 Proof of Theorem 4.1

[Set disjointness communication lower bound (Yao, 1979)] Suppose Alice and Bob are given inputs $a, b \in \{0,1\}^n$, respectively, with the goal of jointly computing $\mathrm{DISJ}(a,b) = \max_i a_i b_i$ by alternately sending a single bit message to the other party over a sequence of communication rounds. Any deterministic protocol for computing $\mathrm{DISJ}(a,b)$ requires at least n rounds of communication.

$$
r_i = \begin{cases} 0 & \text{if } \exists\, a, b \; s.t. \; g(x_i, x_a, x_b) = 0 \\ 1 & \text{else} \end{cases}
$$

In normal cases, judging the value of r_i requires calculating $g(x_i, x_a, x_b)$ for all $a \in [0, n_d)$ and $b \in [n_d, n_d + n_q)$. Here we simplify the question, and we consider the situation where $g(x_i, x_a, x_b) = 0$ only if $b = a + n_d$ and $n_q = n_d$. Apparently, this is a special case of the original problem, and if one layer of self-attention fail to solve this, it is impossible for it to solve the original problem.

If we assume that the input is like,

$$\boldsymbol{x}_i \in \begin{cases} \{\boldsymbol{x}_i\} & \text{if } i = 0, \\ \{0, \boldsymbol{x}_a\} & \text{if } i \in \{1, \ldots, n_d - 1\}, \\ \{0, \boldsymbol{x}_b\} & \text{if } i \in \{n_d, \ldots, 2 \cdot n_d - 1\}. \end{cases} \tag{6}$$

Given input $(a, b) \in \{0,1\}^{n_d} \times \{0,1\}^{n_d}$, let $x_i = x_a$ if and only if $a_i = 1$ and let $x_i = x_b$ if and only if $b_{i-n_d} = 1$. In this way $r_i = 0$ if and only if $\mathrm{DISJ}(a, b) = 1$.

For simplicity, we use $n = n_d$. Now consider a more complex situation where Alice and Bob each hold a matrix $\boldsymbol{A} \in \mathbb{R}^{n \times d}$, $\boldsymbol{B} \in \mathbb{R}^{n \times d}$. each row of the matrix contains $\boldsymbol{w}$, so $d = H(\boldsymbol{w})$. and we assume that $\boldsymbol{x} = [\boldsymbol{s}, \boldsymbol{w}]$.

Then

$$\boldsymbol{x}_i = \begin{cases} \boldsymbol{x}_i & \text{if } i = 0, \\ [\boldsymbol{s}, \boldsymbol{a}_i] & \text{if } i \in \{1, \ldots, n_d - 1\}, \\ [\boldsymbol{s}, \boldsymbol{b}_i] & \text{if } i \in \{n_d, \ldots, 2 \cdot n_d - 1\}. \end{cases} \tag{7}$$

Let $\mathrm{DISJ1}(\boldsymbol{A}, \boldsymbol{B}) = \max_i(g'(\boldsymbol{x}, \boldsymbol{a}_i, \boldsymbol{b}_i))$, where the function g' acts similar with g, but it takes $\boldsymbol{a}, \boldsymbol{b}$ as input, and $\boldsymbol{x}$ is a fixed constant as $\boldsymbol{x}_i$. So the calculation of $DISJ1$ requires $n \times H(\boldsymbol{w})$ bits of communication.

Also similar to the setting of x, $r_i = 1$ if and only if $\mathrm{DISJ1}(\boldsymbol{A}, \boldsymbol{B}) = 1$.

Then, this is the same with the 3Match problem, following the proof of Theorem 7 in Sanford et al. (2024), and with the following form of transformer $f(\boldsymbol{X})$, $2pH \log\log n + mpH \log\log n \approx mpH \log\log n$ bits are communicated.

$$f(\boldsymbol{X}) = \phi(f_h(\boldsymbol{X})),$$

where ϕ stands for the feed forward layers.

$$f_h(\boldsymbol{X}) = \frac{\sum_{i=1}^{N} \exp\left((\boldsymbol{W}_q x_1)^T \boldsymbol{W}_k x_i\right) \boldsymbol{W}_v x_i}{\sum_{i=1}^{N} \exp\left((\boldsymbol{W}_q x_1)^T \boldsymbol{W}_k x_i\right)}$$

Therefore, only we require $mpH \log\log n \geq nH(\boldsymbol{w}) \rightarrow mph \geq nH(\boldsymbol{w}) / \log\log n$.

Theorem C.1. *For input documents of length n, if $mpH \leq \Omega(nH(\boldsymbol{w}) / \log\log n)$, then there is no one layer transformer $\mathcal{M}$ with embedding size m, precision p and H heads satisfying $\mathcal{M}(X) = r$.*

Also, as shown in Sanford et al. (2024), multiple layers of multi-headed attention are subject to the same impossibility Every multi-layer transformer that computes Match3 must have width, depth, embedding dimension, or bit complexity at least $N^{\Omega(1)}$.

However, one layer of transformer can effectively solve Match2 problem, for RAG, we can simply fix one or some token to represent the information of a document, in this way $g(x_i, x_a, x_b)$ becomes $g(x_i, x_0, x_b)$, if we use the first transformer layer to aggregrate information of x_i and x_0, then use the second layer to gather x_b and judging the relevance by MLP layers, then the problem can be solved.

therefore, $g(x_i, x_a, x_b) = g(x_i, x_a, x_b) = g'(x'_i, x'_b)$. Then the problem becomes a pair-wise problem, and transformer can easily solve it.

Proposition C.2. *Using transformer to solve* $r_i = \begin{cases} 0 & \text{if } \exists\, a, b \text{ s.t. } g'(x'_i, x'_b) = 0 \\ 1 & \text{else} \end{cases}$ *requires* $mpHD \geq n_d H(\boldsymbol{w})$.

This mainly dues to the self regression of LLM, token x_i can not gather information of token x_j if $j > i$, which means that all the judgment of is done during the calculation of query rather than the document tokens because we put the documents ahead of the query. However, if we put the query first, then $g'(x_i, x_b)$ can be effectively calculated in the document token.

D Proof of Section 5

D.1 Proof of Theorem 5.1

Let $attn(x_i) = (\boldsymbol{W}_q\boldsymbol{x}_i)^T\boldsymbol{W}_k\boldsymbol{X}_{:i}$ be the original attention layer of LLM, and $attn'(x_i) = ((\boldsymbol{W}_q + \Delta\boldsymbol{W}_q)\boldsymbol{x}_i)^T(\boldsymbol{W}_k + \Delta\boldsymbol{W}_k)\boldsymbol{X}_{:i}$ be the finetuned one and $\widehat{attn}$ be desired function, can we finetune the model to be $\widehat{attn}$?

So we need

$$softmax(attn'(\boldsymbol{X}))[i] \approx \begin{cases} 0 & \text{if } \exists\, b \; s.t. \; g_1(x_i, x_b) = 0 \\ softmax(attn(\boldsymbol{X}_r))[i] & \text{else} \end{cases}$$

where $\boldsymbol{X}_r$ means those related tokens. let $\boldsymbol{A}_{i,j} = attn(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{W}_q\boldsymbol{x}_i)^T\boldsymbol{W}_k\boldsymbol{x}_j = \boldsymbol{x}_i^T\boldsymbol{W}\boldsymbol{x}_j$

$$\begin{aligned} attn'(\boldsymbol{x}_i, \boldsymbol{x}_j) &= ((\boldsymbol{W}_q + \Delta\boldsymbol{W}_q)\boldsymbol{x}_i)^T(\boldsymbol{W}_k + \Delta\boldsymbol{W}_k)\boldsymbol{x}_j \\ &= \boldsymbol{x}_i(\boldsymbol{W} + \Delta\boldsymbol{W})\boldsymbol{x}_j \\ &= \boldsymbol{x}_i\boldsymbol{W}\boldsymbol{x}_j + \boldsymbol{x}_i\Delta\boldsymbol{W}\boldsymbol{x}_j \end{aligned}$$

where $\Delta\boldsymbol{W} = \Delta\boldsymbol{W}_q\boldsymbol{W}_k + \boldsymbol{W}_q\Delta\boldsymbol{W}_k + \Delta\boldsymbol{W}_q\Delta\boldsymbol{W}_k$

if there exists $\Delta\boldsymbol{W}_q', \Delta\boldsymbol{W}_k'$ satisfying

$$\boldsymbol{x}_i\Delta\boldsymbol{W}\boldsymbol{x}_j \begin{cases} \leq c_l & \text{if } x_j \text{ is noise} \\ \in [c_h, c_h + \xi_r] & \text{else} \end{cases} \tag{8}$$

if $\boldsymbol{x}_j$ is a noise token and $\boldsymbol{x}_i\Delta\boldsymbol{W}\boldsymbol{x}_j = c_l$, then

$$\begin{aligned} softmax(attn'(\boldsymbol{x}_i))[j] - 0 &= softmax(A_{i,:} + x_i\Delta\boldsymbol{W}\boldsymbol{X})[j] \\ &= \frac{\exp(A_{i,j} + x_i\Delta\boldsymbol{W}x_j)}{\sum_k \exp(A_{i,k} + \boldsymbol{x}_i\Delta\boldsymbol{W}\boldsymbol{x}_k)} \\ &= \frac{\exp(A_{i,j} + c_l)}{\sum_k \exp(A_{i,k} + \boldsymbol{x}_i\Delta\boldsymbol{W}\boldsymbol{x}_k)} \\ &= \frac{\exp(A_{i,j} + c_l - c_h)}{\sum_k \exp(A_{i,k} + \boldsymbol{x}_i\Delta\boldsymbol{W}\boldsymbol{x}_k - c_h)} \end{aligned}$$

if we need $softmax(attn'(\boldsymbol{x}_i))[j] - 0 \leq \epsilon^2$, due to the optimal value is 0, so to ϵ approximate the value, we use ϵ^2, and let $A_{i,j} = \max(A_{i,:})$, then

$$\frac{\exp(A_{i,j} + c_l - c_h)}{\sum_k \exp(A_{i,k} + \boldsymbol{x}_i\Delta\boldsymbol{W}\boldsymbol{x}_k - c_h)} \leq softmax(attn'(\boldsymbol{x}_i))[j] - 0 \leq \epsilon^2$$

$$\exp(A_{i,j} + c_l - c_h) \leq \epsilon^2 \sum_k \exp(A_{i,k} + \boldsymbol{x}_i\Delta\boldsymbol{W}\boldsymbol{x}_k - c_h)$$

$$c_l - c_h \leq \ln\left(\epsilon^2 \sum_k \exp(A_{i,k} + \boldsymbol{x}_i\Delta\boldsymbol{W}\boldsymbol{x}_k - c_h)\right) - A_{i,j}$$

$$c_l - c_h \leq \ln\left(\epsilon^2 n \exp(A_{i,j} + \xi_r)\right) - A_{i,j}$$

$$c_l - c_h \leq \xi_r + \ln \epsilon^2 n$$

else if $\boldsymbol{x}_j$ is a relevant token, let $c = c_h$, and $\sum_{k'} \exp(A_{i,k'})$ denotes the summation of all relevant tokens, we consider a simple case where $\boldsymbol{x}_i\Delta\boldsymbol{W}\boldsymbol{x}_j = c_h$, and for one token $\boldsymbol{x}_{k1}$, we have $\boldsymbol{x}_i\Delta\boldsymbol{W}\boldsymbol{x}_{k1} = c_h + \xi_r$, and for all other relevant tokens we have $\boldsymbol{x}_i\Delta\boldsymbol{W}\boldsymbol{x}_k = c_h$

$$
\begin{aligned}
& softmax(\hat{attn}(\boldsymbol{x}_i))[j] - softmax(attn'(\boldsymbol{x}_i))[j] \\
= & \frac{\exp(A_{i,j})}{\sum_{k'} \exp(A_{i,k'})} - \frac{\exp(A_{i,j} + x_i \Delta \boldsymbol{W} x_j)}{\sum_k \exp(A_{i,k} + x_i \Delta \boldsymbol{W} x_k)} \\
= & \frac{\exp(A_{i,j})}{\sum_{k'} \exp(A_{i,k'})} - \frac{\exp(A_{i,j} + x_i \Delta \boldsymbol{W} x_j - c)}{\sum_k \exp(A_{i,k} + x_i \Delta \boldsymbol{W} x_k - c)} \\
= & \frac{\exp(A_{i,j})}{\sum_{k'} \exp(A_{i,k'})} - \frac{\exp(A_{i,j})}{\sum_k \exp(A_{i,k} + x_i \Delta \boldsymbol{W} x_k - c)}
\end{aligned} \tag{9}
$$

considering $softmax(\hat{attn}(\boldsymbol{x}_i))[j] - softmax(attn'(\boldsymbol{x}_i))[j] = c(\frac{1}{a} - \frac{1}{b}) \leq \epsilon \frac{c}{a}$, $c = \exp(A_{i,j})$, $a = \sum_{k'} \exp(A_{i,k'})$, $b = \sum_k \exp(A_{i,k} + x_i \Delta \boldsymbol{W} x_k - c)$

$$
b - a \leq \epsilon b
$$

$$
(1 - \epsilon) b \leq a
$$

$$
b \leq \frac{a}{1 - \epsilon}
$$

$$
\sum_k \exp(A_{i,k} + x_i \Delta \boldsymbol{W} x_k - c) \leq \frac{\sum_{k'} \exp(A_{i,k'})}{1 - \epsilon}
$$

$$
\sum_{k'} \exp(A_{i,k'} + x_i \Delta \boldsymbol{W} x_k - c) \leq \frac{\sum_{k'} \exp(A_{i,k'})}{1 - \epsilon}
$$

$$
\sum_{k' - k1} \exp(A_{i,k'}) + \exp(A_{i,k1} + \xi_r) \leq \frac{\sum_{k'} \exp(A_{i,k'})}{1 - \epsilon}
$$

$$
\exp(A_{i,k1} + \xi_r) \leq \frac{\epsilon}{1 - \epsilon} \sum_{k' - k} \exp(A_{i,k'}) + \frac{\exp(A_{i,k1})}{1 - \epsilon}
$$

$$
\xi_r \leq \ln \left(\frac{\epsilon}{1 - \epsilon} \sum_{k' - k} \exp(A_{i,k'}) + \frac{\exp(A_{i,k1})}{1 - \epsilon} \right) - A_{i,k1}
$$

Considering the case that $\frac{\epsilon}{1-\epsilon} \sum_{k' - k} \exp(A_{i,k'}) \approx 0$, then we need $\xi_r \lesssim \ln \frac{1}{1-\epsilon}$

Theorem D.1. *if there exists* $attn'(x_i, x_j) = \boldsymbol{x}_i(\boldsymbol{W} + \Delta \boldsymbol{W})\boldsymbol{x}_j$ *satisfying* $attn'(x_i, x_j) - \widehat{attn}(x_i, x_j) \leq \epsilon$, *then we need*

$$
\xi_r \lesssim \ln \frac{1}{1 - \epsilon}
$$

$$
c_l - c_h \leq \xi_r + \ln \epsilon^2 n
$$

D.2 Proof of Theorem 5.2

In the following, we make use of the quantities

$$
N_{max}(t) = \max_{\hat{v} \in \hat{\mathcal{V}}} N_{\hat{v}}(t), \qquad N_{min}(t) = \min_{\hat{v} \in \hat{\mathcal{V}}} N_{\hat{v}}(t),
$$

where

$$
N_{\hat{v}}(t) = \sum_{v \in \mathcal{V}} 1\{d(v, \hat{v}) \leq t\}
$$

counts the number of $v \in \mathcal{V}$ within a "distance" t of $\hat{v} \in \hat{\mathcal{V}}$.

Theorem D.2. (Fano's inequality with approximate recovery in Scarlett and Cevher (2019)) *For any random variables* $v, \hat{v}$ *on the finite alphabets* $\mathcal{V}, \hat{\mathcal{V}}$, *we have*

$$
P_e(t) \geq \frac{H(\boldsymbol{v}|\hat{\boldsymbol{v}}) - 1}{\log \frac{|\mathcal{V}|}{N_{max}(t)}}. \tag{10}
$$

$P_e(t) = \Pr(||\boldsymbol{z} - \hat{\boldsymbol{z}}|| > t)$, *where* $\boldsymbol{z}$ *is inferenced by some function and the input is* $\boldsymbol{v}$.

Assume the input to the feed forward layer $\boldsymbol{v}$, the first l layers are used to identify the relevance and the last few layers are used for inference. The output of the first l layers are $\boldsymbol{v}_l$, and the embedding is used to conduct inference and get the result $\boldsymbol{z}$. So we can say that with probability $p \geq \frac{H(\boldsymbol{v}_l|\hat{\boldsymbol{v}})-1}{\log \frac{|\mathcal{S}|}{N_{max}(t)}}$ the resulting embedding fails to be t close to the original one *i.e.,* $d(\boldsymbol{z}, \hat{\boldsymbol{z}}) \leq t$, where $\hat{\boldsymbol{v}}$ stands for the optimal input where all irrelevant information is filtered and all the related information is contained and $\hat{\boldsymbol{z}}$ is the corresponding output..

Assume that $\hat{\boldsymbol{w}}$ are equally distributed to each token which satisfies $I(\boldsymbol{w}_i; \boldsymbol{v}) = I(\boldsymbol{w}_j; \boldsymbol{v})$, therefore, for each document, the model holds the same probability of mistakenly identify its relevance. Let p_{we} stands for the error probability of identify noise tokens, and δ be the percentage of relevant tokens. With probability δp_{we}, the relevant token is mistakenly regarded as irrelevant, and with probability $(1-\delta)p_{we}$ the irrelevant token is mistakenly regarded as relevant. So there are $\frac{\delta(1-p_{we})}{\delta(1-p_{we})+(1-\delta)p_{we}}$ percent of information about the relevant ones. Also p_{we} percent of relevant information and $1-p_{we}$ percent of irrelevant information are discarded, then $I(\boldsymbol{s}; \boldsymbol{v}_l) = ((1-p_{we}) \cdot \delta + p_{we} \cdot (1-\delta))\, I(\boldsymbol{s}; \boldsymbol{v})$ are the left information about inference, among these, $\frac{\delta(1-p_{we})}{\delta(1-p_{we})+(1-\delta)p_{we}}$ are acutally related information, others are noise information.

In this way,

$$
\begin{aligned}
I(\boldsymbol{v}_l; \hat{\boldsymbol{v}}) &= I(\boldsymbol{v}_l; \boldsymbol{s}) \\
&= \frac{\delta(1-p_{we})}{\delta(1-p_{we})+(1-\delta)p_{we}} \cdot ((1-p_{we}) \cdot \delta + p_{we} \cdot (1-\delta))\, I(\boldsymbol{s}; \boldsymbol{v}) \\
&= \delta(1-p_{we}) \cdot I(\boldsymbol{s}; \boldsymbol{v}) \\
&\leq \delta(1 - \frac{H(\boldsymbol{w}|\boldsymbol{v}-1)}{H(\boldsymbol{w})}) \cdot I(\boldsymbol{s}; \boldsymbol{v}) \\
&= \delta(\frac{I(\boldsymbol{w}; \boldsymbol{v})+1}{H(\boldsymbol{w})}) \cdot I(\boldsymbol{s}; \boldsymbol{v})
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
P_e(t) &\geq \frac{H(\boldsymbol{v}|\hat{\boldsymbol{v}})-1}{\log \frac{|\mathcal{V}|}{N_{max}(t)}} = \frac{H(\boldsymbol{v}) - I(\boldsymbol{v}; \hat{\boldsymbol{v}})}{\log \frac{|\mathcal{V}|}{N_{max}(t)}} \\
&\geq \frac{H(\boldsymbol{v}) - g_1(\delta, I(\boldsymbol{w}; \boldsymbol{v})) \cdot I(\boldsymbol{s}; \boldsymbol{v})}{\log \frac{|\mathcal{V}|}{N_{max}(t)}}
\end{aligned}
$$

where $g_1(\delta, I(\boldsymbol{w}; \boldsymbol{v})) = \delta(\frac{I(\boldsymbol{w}; \boldsymbol{v})+1}{H(\boldsymbol{w})})$

So when there is no noise, the inference can be conducted based on those information, then we have

$$
\begin{aligned}
\Pr\left(||z - \hat{z}|| > t\right) &\geq \frac{H(\boldsymbol{v}) - g_1(\delta, I(\boldsymbol{w}; \boldsymbol{v})) \cdot I(\boldsymbol{s}; \boldsymbol{v})}{\log \frac{|\mathcal{V}|}{N_{max}(t)}} \\
\Pr\left(||z - \hat{z}|| \leq t\right) &\leq 1 - \frac{H(\boldsymbol{v}) - g_1(\delta, I(\boldsymbol{w}; \boldsymbol{v})) \cdot I(\boldsymbol{s}; \boldsymbol{v})}{\log \frac{|\mathcal{V}|}{N_{max}(t)}}
\end{aligned}
$$

Considering the noise in the embedding of $\boldsymbol{v}_l$, and the noise would have negative impact on the inference.

Also the extra noise information contained in $\boldsymbol{v}_l$ is

$$
\begin{aligned}
I(\boldsymbol{v}^-; \boldsymbol{v}_l) &= \frac{(1-\delta)p_{we}}{\delta(1-p_{we})+(1-\delta)p_{we}} \cdot ((1-p_{we}) \cdot \delta + p_{we} \cdot (1-\delta))\, I(\boldsymbol{s}; \boldsymbol{v}) \\
&= (1-\delta)p_{we} \cdot I(\boldsymbol{s}; \boldsymbol{v}) \\
&\geq (1-\delta)\frac{H(\boldsymbol{w}|\boldsymbol{v})-1}{H(\boldsymbol{w})} \cdot I(\boldsymbol{s}; \boldsymbol{v})
\end{aligned}
$$

Consider the best case where $I(\boldsymbol{v}^-; \boldsymbol{v}_l) = (1-\delta)\frac{H(\boldsymbol{w}|\boldsymbol{v})-1}{H(\boldsymbol{w})} \cdot I(\boldsymbol{s}; \boldsymbol{v})$

Theorem D.3 (Theorem 2 of Kawaguchi et al. (2023))**.** *Let $\mathcal{D} \subseteq \{1, 2, \ldots, D+1\}$. Then, for any $\delta > 0$, with probability at least $1-\delta$ over the training set s, the following generalization bound holds:*

$$\Delta(s) \leq \min_{l \in \mathcal{D}} Q_l, \tag{11}$$

where for $l \leq D$,

$$Q_l = G_3^l \sqrt{\frac{\left(I(X; Z_l^s|Y) + I(\phi_l^S; S)\right)\ln(2) + \widehat{\mathcal{G}}_2^l}{n}} + \frac{G_1^l(\zeta)}{\sqrt{n}};$$

and for $l = D+1$,

$$Q_l = \mathcal{R}(f^s)\sqrt{\frac{I(\phi_l^S; S)\ln(2) + \check{\mathcal{G}}_2^l}{2n}},$$

Here, $S \sim \mathcal{P}^{\otimes n}$, $G_1^l(\zeta) = \hat{\mathcal{O}}(\sqrt{I(\phi_l^S; S) + 1})$, $\widehat{\mathcal{G}}_2^l = \hat{\mathcal{O}}(1)$, $\check{\mathcal{G}}_2^l = \hat{\mathcal{O}}(1)$, and $G_3^l = \hat{\mathcal{O}}(1)$ as $n \to \infty$. The formulas of $G_1^l(\zeta)$, $\widehat{\mathcal{G}}_2^l$, $\check{\mathcal{G}}_2^l$, and G_3^l are given in Appendix.

using $||f(x) - \hat{f}(x)||$ as the loss function, then we have that

$$\begin{aligned}
\mathcal{L} - \hat{\mathcal{L}} &\leq G_3^l \sqrt{\frac{\left(I(X; Z_l^s|Y) + I(\phi_l^S; S)\right)\ln(2) + \widehat{\mathcal{G}}_2^l}{n}} + \frac{G_1^l(\zeta)}{\sqrt{n}} \\
&= c_1 \sqrt{I(\boldsymbol{v}^-|\boldsymbol{v}_l) + I(\phi_l^S; S)} + c_3 \\
&\leq c_1 \sqrt{I(\boldsymbol{v}^-|\boldsymbol{v}_l)} + c_1 \sqrt{I(\phi_l^S; S)} + c_3 \\
&\leq c_1 \sqrt{I(\boldsymbol{v}^-|\boldsymbol{v}_l)} + c_2
\end{aligned} \tag{12}$$

where $c_2 = c_3 + c_1\sqrt{I(\phi_l^S; S)}$ Therefore,

$$\Pr\left(||f(x) - \hat{f}(x)|| \leq t + c_1\sqrt{I(\boldsymbol{v}^-|\boldsymbol{v}_l)+} + c_2\right) \leq 1 - \frac{H(\boldsymbol{v}) - g_1(\delta, I(\boldsymbol{w}; \boldsymbol{v})) \cdot I(\boldsymbol{s}; \boldsymbol{v})}{\log \frac{|\mathcal{V}|}{N_{max}(t)}} \tag{13}$$

with $I(\boldsymbol{v}^-; \boldsymbol{v}_l) = (1-\delta)\frac{H(\boldsymbol{w}|\boldsymbol{v})-1}{H(\boldsymbol{w})} \cdot I(\boldsymbol{s}; \boldsymbol{v}) = g_2(\delta, I(\boldsymbol{w}; \boldsymbol{v})) \cdot I(\boldsymbol{s}; \boldsymbol{v})$.

$$\begin{aligned}
&\Pr\left(||f(x) - \hat{f}(x)|| > t + c_1\sqrt{g_2(\delta, I(\boldsymbol{w}; \boldsymbol{v})) \cdot I(\boldsymbol{s}; \boldsymbol{v})+} + c_2\right) \\
&\quad > \frac{H(\boldsymbol{v}) - g_1(\delta, I(\boldsymbol{w}; \boldsymbol{v})) \cdot I(\boldsymbol{s}; \boldsymbol{v})}{\log \frac{|\mathcal{V}|}{N_{max}(t)}}
\end{aligned} \tag{14}$$

Theorem D.4. *For a Feed Forward Network f and the input x contains $1-\delta$ percent of noise information, assume the optimal function is $\hat{f}(x)$ which filter out the noise and finish the inference, then*

$$\Pr\left(||f(x) - \hat{f}(x)|| > t'\right) > \frac{H(\boldsymbol{v}) - g_1(\delta, I(\boldsymbol{w}; \boldsymbol{v})) \cdot I(\boldsymbol{s}; \boldsymbol{v})}{\log \frac{|\mathcal{V}|}{N_{max}(t)}}, \tag{15}$$

where $t' = t + c_1\sqrt{g_2(\delta, I(\boldsymbol{w}; \boldsymbol{v})) \cdot I(\boldsymbol{s}; \boldsymbol{v})} + c_2$. $g_1(\delta, I(\boldsymbol{w}; \boldsymbol{v})) = \delta(\frac{I(\boldsymbol{w}; \boldsymbol{v})+1}{H(\boldsymbol{w})})$ $g_2(\delta, I(\boldsymbol{w}; \boldsymbol{v})) = (1-\delta)\frac{H(\boldsymbol{w}|\boldsymbol{v})-1}{H(\boldsymbol{w})}$