MuDAF: Long-Context Multi-Document Attention Focusing through Contrastive Learning on Attention Heads

Anonymous ACL submission

Abstract

Large Language Models (LLMs) frequently show distracted attention due to irrelevant information in the input, which severely impairs their long-context capabilities. Inspired by recent studies on the effectiveness of retrieval heads in long-context factutality, we aim at addressing this distraction issue through improving such retrieval heads directly. We propose **Mu**lti-**D**ocument **A**ttention **F**ocusing (*MuDAF*), a novel method that explicitly optimizes the attention distribution at the head level through contrastive learning. According to the experimental results, *MuDAF* can significantly improve the long-context question answering performance of LLMs, especially in multi-document question answering. Extensive evaluations on retrieval scores and attention visualizations show that *MuDAF* possesses great potential in making attention heads more focused on relevant information and reducing attention distractions.

1 Introduction

As large language models (LLMs) continue to advance and find broader applications, the demand for their ability to efficiently handle ultra-long texts is growing. For instance, in Retrieval-Augmented Generation (RAG) systems (Gao et al., 2023; Jin et al., 2024) and LLM agent systems (Guo et al., 2024), models are often required to extract critical information from long-text corpora to accomplish complex generative tasks. However, research has shown that the real context window size of existing models often falls short of their claimed capabilities (An et al., 2024a), revealing significant shortcomings in their ability to utilize information from long inputs. This issue is particularly evident in two major challenges: the "lost-in-the-middle" (Liu et al., 2024a) phenomenon, where the middle portions of the text are neglected, and the interference from irrelevant information (Shi et al., 2023; Wu et al., 2024a). These challenges substantially hinder the performance of models in long-context tasks.

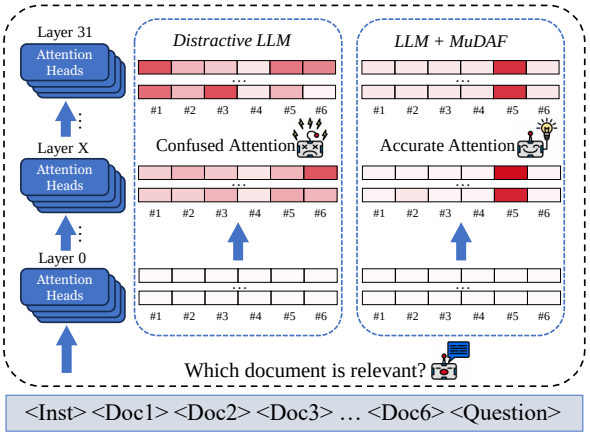

Figure 1: Given instructions, long documents and a specific question, LLMs can often be confused when facing information from multiple sources. Our method *MuDAF* helps LLMs focus on documents related to the given question. Deeper colors represent higher attention values.

In recent years, many studies have conducted in-depth analyses of the role of attention mechanisms in long-context modeling (Chen et al., 2024; Hong et al., 2024; Zheng et al., 2024). Notably, some research has identified that specific attention heads found in the **Needle-in-a-Haystack (NIAH)** test are critical for long-context factuality and has named them retrieval heads (Wu et al., 2024c), which can perform a copy-paste operation from the input context to the output. Inspired by these works, we are extremely curious about such a question: *How can we strengthen these retrieval heads directly to enhance models' long-context modeling capabilities?*

In this work, we care about **long-context question answering (LCQA)**, especially **multi-document question answering (MDQA)**, where the long input context contains many irrelevant

documents and causes distractions. However, the retrieval heads in MDQA might be different from those in the NIAH test, since the NIAH test only shows the effectiveness of these heads in the copy-paste pattern. Considering the gap between them, we need to identify the retrieval heads in the MDQA setting and prove their effectiveness in helping models utilize relevant information in the context. As expected, we indeed found some retrieval heads that were different from those found in the NIAH test (§3.1), which may point to the fact that attention heads exhibit different levels of retrieval capabilities when applied to different tasks. Then, we explored methods to improve such retrieval heads in MDQA. As we know that attention weights are calculated by the softmax of the scaled dot product between query and key projections (Vaswani, 2017), it's feasible to optimize the attention weights allocation by learning better projections. Therefore, we propose **Mu**lti-**D**ocument **A**ttention **F**ocusing (***MuDAF***), a method that applies contrastive learning on attention heads to help them learn better query-key projections, thus optimizing their attention distributions. As depicted in Figure 1, *MuDAF* can help attention heads be more focused on relevant passages while minimizing interference from irrelevant content.

In summary, our contributions are as follows.

- We provide a method to assess the retrieval capabilities of attention heads in multi-document question answering, distinguishing special retrieval heads that are different from those found in the NIAH test.
- We propose *MuDAF*, a novel approach based on contrastive learning that optimizes the attention pattern at the head level to improve the long-context modeling ability of LLMs, especially in MDQA tasks.
- Experiments show that our methods can significantly enhance the long-context performance of LLMs and surpass GPT-4o in some datasets.
- We did further analysis and ablations to show the effectiveness of our methods. Providing several insights about enhancing retrieval heads in MDQA.

2 Related Work

Attention-Based Salience for Long-Context. Since the attention mechanism was first introduced by (Bahdanau, 2014), attention weight has become an important tool for interpreting important information in the input sequence (Serrano and Smith, 2019; Ferrando et al., 2024). For example, Peysakhovich and Lerer (2023) use attention weights to estimate the importance of documents that can be leveraged to arrange their positions, thus improving the performance of long-context LLMs. Xiao et al. (2024) manage to reduce KV cache for attention heads based on their attention patterns. He et al. (2024a) investigate the importance of attention weights in knowledge retention. Obviously, the attention mechanism has not only been a critical and reliable information resource for processing various long-context tasks (Xiao et al.; Chen et al., 2024) but also presented substantial potential for further exploration and optimization (Wu et al., 2024c; Lu et al., 2024; He et al., 2024b). Our approach also highlights the function of certain attention heads in in-context retrieval (Ram et al., 2023), aiming at optimizing attention distribution to get better long-context LLMs.

Distractions by Irrelevant Content. Previous research has shown that LLMs can be easily disturbed by irrelevant context (Shi et al., 2023; Wu et al., 2024b), making them overallocate attention to useless content. Some methods have been proposed to mitigate such issues. Liu et al. (2024b) introduce an innovative framework that helps LLMs recognize relevant entities in long contexts through efficient reference management. Wu et al. (2024d) reduce distractions by aligning the representations of the original context and the retrieved sub-context. Xiong et al. (2024) enhance the retrieval capabilities of LLMs in highly similar contexts through fine-tuning on synthetic data. Another method proposes a novel differential attention mechanism to amplify attention to the relevant context while canceling attention noise (Ye et al., 2024). However, these methods do not explicitly optimize the attention distribution based on the input context, while our method provides a more straightforward and effective way.

Contrastive Learning on Generative Models. As a self-supervised training technique, contrastive learning (cho, 2005; Hadsell et al., 2006; Robinson et al., 2021) has been widely leveraged in NLP tasks such as sentence embedding (Gao et al., 2021). With the advancement of generative language models (Radford et al., 2019), contrastive learning has also exhibited great potential in decoder-only architectures to achieve bet-

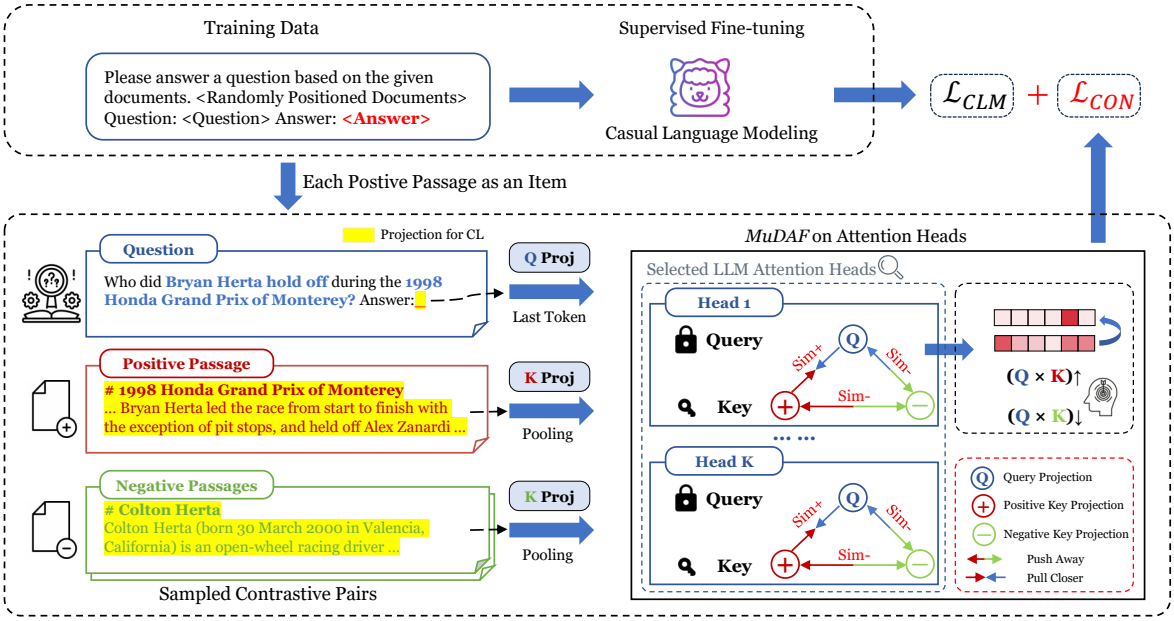

Figure 2: An overview of our proposed method. The goal of *MuDAF* is to adjust the similarity between the `Query` features from the question and the `Key` features from the passages, thus making attention heads allocate more attention weights in relevant information and reducing distractions. CL means contrastive learning.

ter hidden expressiveness (Su et al., 2022; Jain et al., 2023; Yan et al., 2024). For long-context tasks, Caciularu et al. (2022) utilize contrastive learning to explicitly discriminate representations of supporting evidence sentences from negative ones in long-context QA. Wu et al. (2024d) also leverage contrastive learning to align representations of different contexts. However, our method applies contrastive learning inside the attention head components instead of sequence representations. To the best of our knowledge, we are the first to show the effectiveness of optimizing attention distributions by adjusting the similarity between query and key projections at the head level directly.

3 Method

In this section, we introduce our proposed method. We start by investigating the relationship between the performance of an LLM in MDQA and the ability of its attention heads for information retrieval to identify its retrieval heads (§3.1). We then discuss the details of our method (§3.2). An overview of our approach is provided in Figure 2.

3.1 Attention Heads Responsible for IR

Information retrieval (IR) here means recognizing required information from noisy input context. Wu et al. (2024c) have proven the existence of retrieval heads in the NIAH test (i.e., NIAH retrieval heads), which implement the conditional copy algorithm and redirect information from the input to the output. However, it remains unclear whether these retrieval heads function similarly in other long-context tasks, such as MDQA, where LLMs are required to retrieve relevant information from previous passages to answer a given question rather than simply repeating patterns found in the context.

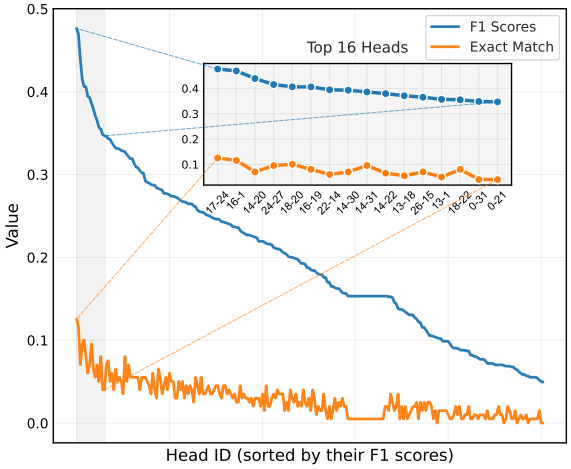

Figure 3: The F1 and EM retrieval scores for attention heads of Llama3.1-8B. We list top 16 retrieval heads ranked by their F1 scores in the inner graph.

First, we manually labeled golden passages for all questions in the HotpotQA subset of LongBench. Our annotation pipeline can be found in Ap-

pendix A.1. We define the retrieval score of a given attention head as their ability to attend to golden passages among all input passages. Formally, for each question q, we have several relevant golden passages P_G and quite a few irrelevant passages P_I. We mixed all P_G and P_I into a long input context $\mathcal{C}$ in random order, then we concatenated the question q to the end of $\mathcal{C}$ to obtain the final prompt $\mathcal{P}$. For each attention head, we calculated its attention score over all input passages by summing the attention scores between the last token of the input and all tokens in the corresponding passages. The passage whose attention score was higher than a given threshold ϵ would be considered an attended passage P_{A_h} of attention head h. We then calculated the F1 score and EM score based on P_G, P_I and P_A. Figure 3 presents the curves of F1 scores and EM scores of all attention heads in Llama-3.1-8B (Meta, 2024), ranked in descending order of F1 scores. The final retrieval score $\mathcal{R}_h$ ($0 \leq \mathcal{R}_h \leq 1$) of an attention head h is the average F1 score on all HotpotQA test cases. The formula for calculating $\mathcal{R}_h$ is as follows:

$$\mathcal{R}_h = mean(F1\_Score(P_G, P_I, P_A)) \quad (1)$$

Details about the calculation process can be found in Appendix A.2.

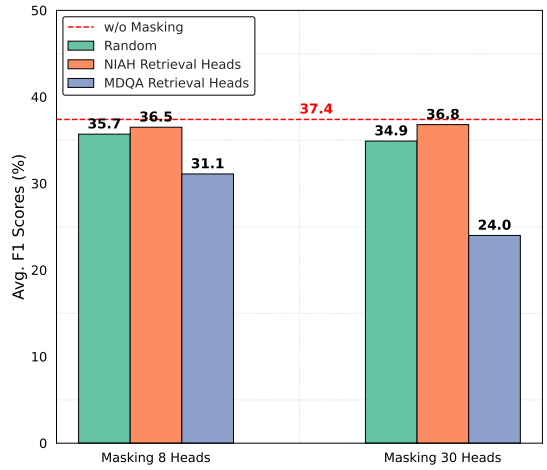

Figure 4: Average performance of Llama3.1-8B on LongBench with different masking strategies. In this experiment, we used the MDQA subset of LongBench, including HotpotQA, 2WikiMQA and MuSiQue. Masked retrieval heads were also randomly selected from the set of retrieval heads, and the final results were obtained by averaging three independent experimental runs.

We found that retrieval scores decline smoothly from the strongest heads to the weakest heads, making it difficult to distinctly classify them as either "strong" or "weak" based on a clear threshold. For convenience, **we consider the top 50 attention heads (about 5% of the total) as retrieval heads**. To confirm the essentiality of these retrieval heads, we then carried out further masking experiments. As shown in Figure 4, the performance on LongBench is severely damaged when strong MDQA retrieval heads are masked, showing that they play a vital role in multi-document modeling. In addition, the model's performance exhibits smaller fluctuations when random attention heads or NIAH retrieval heads are masked. This experimental result gives us a reliable direction for selecting which attention heads to optimize. In other words, we may assume that only by optimizing attention heads with a strong enough retrieval capability can we improve the long-context modeling ability of the model.

3.2 Contrastive Learning for Optimizing Attention Heads

We have proven that MDQA retrieval heads can offer reliable key information in input context, helping LLMs leverage golden information to answer the given question. Therefore, we propose *MuDAF*, a method based on joint training of casual language modeling (CLM) and contrastive learning, aiming at enhancing MDQA retrieval heads for a better focus on relevant context and reducing the distraction caused by irrelevant content.

Preliminary. To better describe our method, we first define some universal notations and variables:

- $\mathcal{H}$: attention head set of the model. $\mathcal{H}_i$ means the attention head set in the ith layer of the model.
- N: the number of layers that the model has.
- k and $\mathcal{K}$: k indicates the index of a certain passage. $\mathcal{K}$ means the number of passages for a given example.
- $[h]$: a superscript that denotes a specific attention head.

Attention Simplification for MDQA. We perform a simplification to the attention mechanism in the MDQA setting that makes it easy to understand our optimization goal.

Let $\mathcal{C}$ represent the holistic long context, which consists of golden passages P_G, irrelevant passages P_I and the question q. We assume that LLMs can better answer a question if they attain more hidden information directly from golden passages P_G.

To describe this information aggregation process more concisely, we define the hidden information of passage P_k stored at layer i as $\mathcal{I}_{i,k}$, and we only consider the attention distribution of the last input token. We simply define a substitutive attention weight of head h for a passage P_k as $\mathcal{A}_k^{[h]}$. Meanwhile, for other tokens in the input, we define their attention weight and hidden information as $\mathcal{A}_\circ^{[h]}$ and $\mathcal{I}_{i-1}$. Then we can simply denote the information obtained by layer i as follows:

$$\mathcal{I}_i = \mathcal{O}_i\Bigg(\text{concat}_h^{\mathcal{H}_i}\Big\{\sum_k^{\mathcal{K}} \mathcal{A}_k^{[h]} \cdot \mathcal{I}_{i-1,k} + \mathcal{A}_\circ^{[h]}\mathcal{I}_{i-1}\Big\}\Bigg) \tag{2}$$

where $\mathcal{O}_i$ represents the output projection module of layer i, $\mathcal{H}_i$ represents the attention heads set of layer i, $\mathcal{A}_k^{[h]}$ is the attention score on passage P_k, while $\mathcal{A}_\circ^{[h]}$ represents the sum of attention scores of all other non-passage tokens. To make LLMs gain more information from golden passages and mitigate the disturbance by irrelevant passages, our optimization goal is to increase the attention weights assigned to golden passages P_G while reducing the attention weights assigned to irrelevant passages P_I. Formally, a multi-head attention score (Vaswani, 2017) between the last input token t and a token i inside a passage can be expressed as:

$$\begin{aligned}\text{attn}_i^{[h]} &= \left[\text{softmax}\left(\frac{Q_t^{[h]}(K^{[h]})^T}{\sqrt{d}}\right)\right]_i \\ &= \frac{\exp\left(\frac{Q_t^{[h]}(K_i^{[h]})^T}{\sqrt{d}}\right)}{\sum_j \exp\left(\frac{Q_t^{[h]}(K_j^{[h]})^T}{\sqrt{d}}\right)}.\end{aligned} \tag{3}$$

Intuitively, the attention score can be raised through aligning the representations of $Q_t^{[h]}$ and $K_i^{[h]}$, as well as pushing apart $Q_t^{[h]}$ and K projections of other tokens in the embedding space. Therefore, we can leverage contrastive learning (Hadsell et al., 2006) at the head level inside the attention mechanism to adjust the attention allocation. In our approach, we argue that aggregating the representations of all tokens in a passage for contrastive learning is as effective as performing contrastive learning on the representation of each individual token. Hence, we perform an average pooling operation on K projections of all tokens to obtain the overall K representation of a passage. Then the attention weight $\mathcal{A}_k^{[h]}$ can be written as:

$$\mathcal{A}_k^{[h]} = \sum_{t' \in P_k} \text{attn}_{t'}^{[h]} \approx \text{attn}_{P_k}^{[h]} \tag{4}$$

$$\text{attn}_{P_k}^{[h]} = \text{softmax}\left(\frac{Q_t^{[h]}\left(\frac{1}{|P_k|}\sum_{t' \in P_k} K_{t'}^{[h]}\right)^T}{\sqrt{d}}\right) \tag{5}$$

We denote the pooled K representation of passage P_k as K_P:

$$K_P^{[h]} = \frac{1}{|P_k|}\sum_{t' \in P_k} K_{t'}^{[h]} \tag{6}$$

Objective of Contrastive Learning. To magnify the attention weight allocated to golden passages, the objective of the contrastive learning is to maximize the similarity between $Q_t^{[h]}$ and $K_{P_G}^{[h]}$, while pushing apart the representations of $Q_t^{[h]}$ and $K_{P_I}^{[h]}$. Therefore, the loss function can be presented as follows:

$$\mathcal{L}_{\text{CON}} = -\sum_h \log \frac{e^{(\text{sim}(Q_t^{[h]}, K_{P_k}^{[h]})/\tau)}}{\sum_{P_j \in P} e^{(\text{sim}(Q_t^{[h]}, K_{P_j}^{[h]})/\tau)}}. \tag{7}$$

where $P = P_G \cup P_I$, $\text{sim}(\cdot,\cdot)$ denotes the cosine similarity function and τ is a temperature hyperparameter. Finally, we combine L_{CON} with a Causal Language Modeling (CLM) loss function as the overall loss function:

$$\mathcal{L} = \mathcal{L}_{\text{CLM}} + \lambda\mathcal{L}_{\text{CON}} \tag{8}$$

where λ is a hyperparameter to control the weight of $\mathcal{L}_{\text{CON}}$.

4 Experiments

4.1 Setup

Benchmarks We evaluated our fine-tuned models on LCQA datasets, including both multi-document question answering and single-document question answering subsets from LongBench (Bai et al., 2024) and ZeroSCROLLS benchmarks (Shaham et al., 2023). LongBench is a bilingual and multitask benchmark for long-context understanding. Among its subsets, we included HotpotQA (Yang et al., 2018), 2WikiMQA (Ho et al., 2020), MuSiQue (Trivedi et al., 2022) and Qasper (Dasigi et al., 2021). ZeroSCROLLS also provides various datasets for evaluating models'

Dataset	Type	Avg #Words	#Items
LongBench			
HotpotQA	Multi-Doc QA	9,151	200
2WikiMQA	Multi-Doc QA	4,887	200
MuSiQue	Multi-Doc QA	11,214	200
Qasper	Single-Doc QA	3,619	200
ZeroSCROLLS			
MuSiQue	Multi-Doc QA	1,749	500
Qasper	Single-Doc QA	3,531	500

Table 1: An overview of the benchmark statistics. The metric for these datasets are all F1 scores. Note that the MuSiQue subset in ZeroSCROLLS contains unanswerable questions and models should refused to answer them.

capabilities in synthesizing information over long texts, and we included the MuSiQue and Qasper subsets from it. The statistics of benchmarks are listed in Table 1.

Baselines and Foundation Models We compared our approach to several popular and strong long-context LLMs, including GPT-3.5 Turbo (OpenAI, 2022), GPT-4o (OpenAI, 2024), FILM-7B (An et al., 2024b), ChatQA-2-8B (Xu et al., 2024), ProLong-8B-64k[1] (Gao et al., 2024), and Llama3.1-8B-Instruct-128k (Meta, 2024). In this paper, our method are applied to the foundation model Llama-3.1-8B (Dubey et al., 2024), which has 128K context length. We also include an intuitive vanilla supervised fine-tuning (*Vanilla-SFT*) baseline that fine-tuning the foundation model only with $\mathcal{L}_{\text{CLM}}$ loss.

Training Data We adopted the training dataset of HotpotQA (Yang et al., 2018) with additional hard negative passages. Here negative samples are enriched by similar passages collected from the work by Jiang et al. (2024), where they grouped multiple Wikipedia documents through hyperlinks. Thus, we expanded negative passages utilizing the group of existing passages in the original set.

4.2 Implementation Details

Target Heads Selection To implement *MuDAF*, we first select several strong MDQA retrieval heads for contrastive learning. Since we have got retrieval scores of all attention heads in section §3.1, we could simply select retrieval heads with the highest retrieval score. However, we found that this greedy strategy is not the best and not robust. We thus use a weighted random selection algorithm that randomly picks attention heads based on their retrieval scores. More specifically, given attention heads set $\mathcal{H}$ and their retrieval scores $\{\mathcal{R}_h\}$, the probability $P(h)$ of selecting the attention head h is computed as:

$$P(h) = \frac{e^{\mathcal{R}_h/\tau}}{\sum_{h' \in \mathcal{H}} e^{R_{h'}/\tau}}$$

where $\tau > 0$ is a temperature parameter (e.g., 0.05), $\mathcal{H}$ denotes all attention heads. We set the number of selected target heads to 8.

Fine-tuning Details One training sample of contrastive learning contains one golden passage to the given question and many negative passages. We separately compute Q projection for the last token of the question and pooled K_P projection for all passages. During the similarity calculation, we concatenate corresponding representations from all selected attention heads and calculate the overall cosine similarity between them. We found that this implementation is more stable and also effective compared with calculating the similarity for each attention head separately. For *Vanilla SFT*, the order of input passages is randomly shuffled before forming an MDQA input and computing CLM loss. More details can be found in Appendix B.

4.3 Main Results

Enhancement on LCQA Performance. Our method significantly enhances the model's LCQA performance. Table 2 compares the performance of our method with other baselines. *MuDAF* shows great potential in enhancing the LCQA performance of models, getting **+12.7%** improvement on average scores compared with the *Vanilla-SFT* baseline. Meanwhile, our method is also effective on single-document question-answering datasets (e.g., Qasper), indicating that our method is also robust in enhancing the retrieval capabilities of LLMs in one long document. Moreover, our method achieves comparable performance to that of GPT-4o, and even performs better on some datasets, proving the effectiveness of our method. Note that one in five questions of the MuSiQue subset from ZeroSCROLLS are unanswerable, which may have affected the performance of our method on this dataset.

Achieving More Focused Retrieval Heads. Besides improvements on QA performance, we

[1] https://huggingface.co/princeton-nlp/Llama-3-8B-ProLong-64k-Base

Models	LongBench				ZeroSCROLLS		Avg.
	HotpotQA	2WikiMultihopQA	MuSiQue	Qasper	MuSiQue	Qasper	
GPT-4o (OpenAI, 2024)	68.3	49.5	39.8	46.1	59.5	48.7	52.0
GPT-3.5-Turbo (OpenAI, 2022)	51.6	37.7	26.9	43.3	52.0	27.1	39.8
FILM-7B (An et al., 2024b)	62.1	47.0	39.0	42.2	**35.2**	**54.7**	46.7
ChatQA-2-8B (Xu et al., 2024)	52.5	41.8	38.9	28.5	27.3	47.9	39.5
ProLong-8B-64k (Gao et al., 2024)	43.0	24.9	21.3	27.8	25.7	36.7	29.9
Llama3.1-8B-Instruct (Meta, 2024)	54.7	44.0	32.8	**44.7**	29.1	51.8	42.8
Llama3.1-8B-*Vanilla-SFT*	46.8	50.5	28.9	29.2	30.1	41.6	37.8
Llama3.1-8B-*MuDAF-weak*	62.5	53.8	43.1	34.9	23.2	41.8	43.2
Llama3.1-8B-*MuDAF* **(ours)**	**69.6**	**66.2**	**48.2**	40.0	31.2	47.9	**50.5**

Table 2: F1 scores (%) on all tested datasets. Underlined numbers denotes the best performance among all listed models. **Bold** numbers indicates the best performance of tested open source models and our models. *Vanilla-SFT* means training the foundation model without $\mathcal{L}_{\text{CON}}$. *MuDAF-weak* means that we apply our method to weak attention heads (§4.4). *MuDAF* achieves better performance among all tested datasets compared with the *Vanilla-SFT* baseline.

want to examine whether *MuDAF* can genuinely strengthen the retrieval capabilities of target attention heads. Therefore, we make a comparison of target attention heads' retrieval scores between the original model, *MuDAF* and *Vanilla-SFT*. As shown in Figure 5, all target attention heads get significant enhancement from *MuDAF* on retrieval scores compared with the original model, and the optimization gains are also substantially larger than those observed with the *Vanilla-SFT* baseline. For example, head 16-9 achieves a **+0.48** improvement in retrieval scores through *MuDAF*, elevating its ranking from **the 119th to the 3rd** place instantaneously. In contrast, the *Vanilla-SFT* baseline brings few improvements to its ranking. It is worth noting that stronger heads often achieve smaller improvements, indicating that it is easier to enhance those attention heads in the middle part.

4.4 Analysis

Effectiveness on Weak Heads. In this paper, we regard attention heads with low retrieval scores (i.e., $\mathcal{R}_h < 0.1$) as weak attention heads. Due to the promising improvements of both performance and retrieval scores when applying *MuDAF* to strong retrieval heads, we are curious about whether our method could transform weak attention heads into heads with a certain retrieval capability. Therefore, we randomly selected attention heads whose retrieval scores are nearly zero for optimization. As shown in Table 2, the overall performance is relatively weak, but it is still an improvement compared with the *Vanilla-SFT* baseline. We further calculated their retrieval scores after the training stage. As illustrated in Figure 6, *MuDAF* can exactly enhance the retrieval capabilities of these weak heads, while *Vanilla-SFT* does nothing in it. This phenomenon manifests that we could adjust the attention pattern of one head through *MuDAF* towards retrieval heads even though they are extremely weak attention heads in the original model. But at the same time, obviously we can hardly achieve the same performance as strong retrieval heads do since their attention values are still relatively low in the middle context.

Whole-layer Optimization. Although we are focusing on target attention heads in previous experiments, the model parameters of other parts are not frozen, making them possible to get improved as well. So we also calculated the attention scores after the training for other attention heads that are not directly selected for the contrastive learning. Surprisingly, we found that most attention heads within the same layer can also be optimized when incorporating at least one attention head in the contrastive learning process. We discuss more about this interesting phenomenon in Appendix C.

Bottleneck When Scaling the Number of Target Heads. We wonder if it is possible to get a stronger model through applying *MuDAF* to more attention heads. Unfortunately, we discovered a bottleneck when scaling the number of trained attention heads. As depicted in Figure 7, we did not see much improvement if we consistently increased the number of selected attention heads beyond 8 heads. Furthermore, if we engage all attention heads in contrastive learning, the training will become unstable and struggle to converge, leading to a collapse of the overall performance. We re-

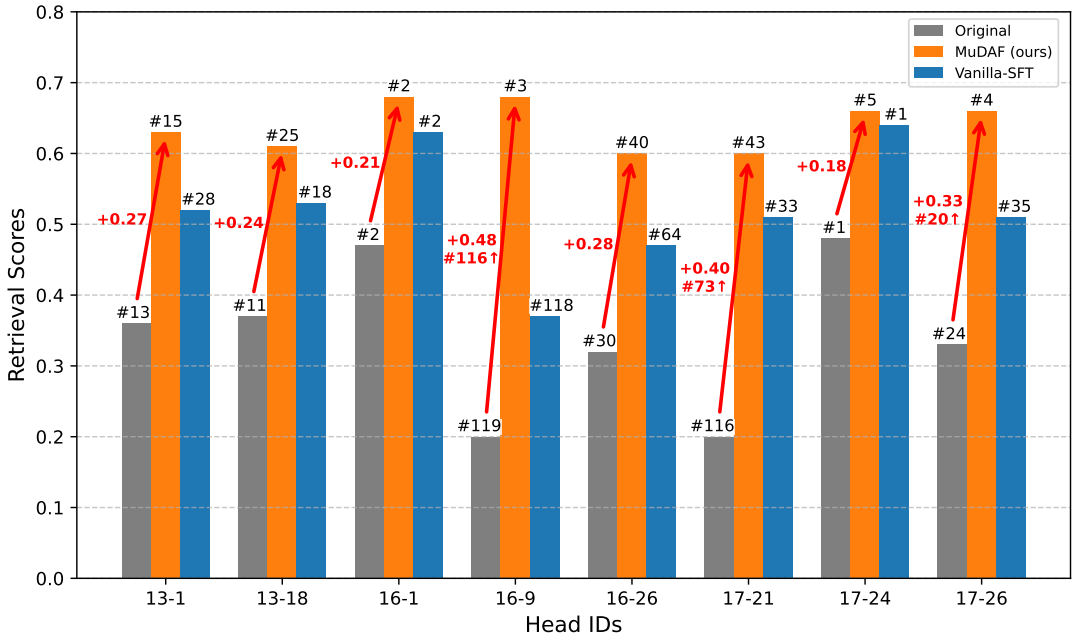

Figure 5: Comparison of enhanced retrieval capabilities in selected attention heads. We annotate the rank of each attention head among all heads above the bar (i.e., #x).

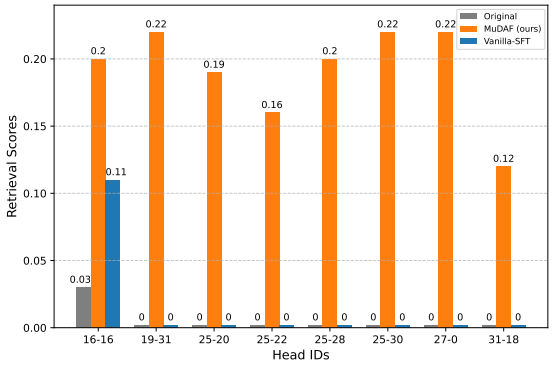

Figure 6: Enhanced retrieval capabilities when applying *MuDAF* to weak attention heads.

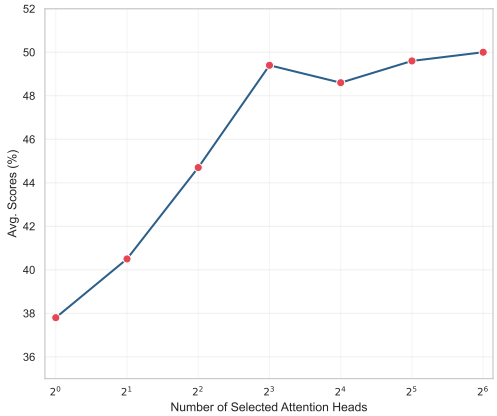

Figure 7: Ablation on the number of selected attention heads. In this ablation, we select attention heads according to the retrieval score from high to low.

main the investigation of this bottleneck as future research.

Case study. We present a case study to show the effectiveness of *MuDAF* to make the model be more focused on relevant passages within a long context. Through our observations, our approach can correct incorrect attention assignments while enhancing attention weights allocated to golden documents, exhibiting its potential to convert weak attention heads into retrieval-capable attention heads. Details are provided in Appendix E.

5 Conclusion

In this paper, we focus on optimizing specific attention heads to enhance their ability to concentrate on relevant content in LCQA tasks. Our analysis reveals the existence of specialized retrieval heads in the MDQA setting that differ from those found in the NIAH test. To improve these retrieval heads, we introduce *MuDAF*, an approach that significantly enhances the retrieval capabilities of attention heads in MDQA regardless of their initial strength. Consequently, the performance of LLMs in LCQA tasks gets remarkable improvements as well. Our method and experiments draw a promising roadmap and provide valuable insights in utilizing contrastive learning to optimize the attention distribution at the head level in MDQA tasks.

Limitations

Although we see improvements on attention scores and the LCQA performance through our method, it is still hard to explain the relationship between optimizing a certain head's attention distribution and the final output of the model, since other attention heads also engage in reasoning and making the final response.

Moreover, our approach can be affected by the position of the question, which means the model can better retrieve relevant documents in the input through attention focusing only when the question is at the end of the input sequence. It is of great importance if we can design a more robust method to mitigate such positional bias.

References

2005. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 539–546. IEEE.

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. 2023. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4895–4901.

Chenxin An, Jun Zhang, Ming Zhong, Lei Li, Shansan Gong, Yao Luo, Jingjing Xu, and Lingpeng Kong. 2024a. Why does the effective context length of llms fall short? *arXiv preprint arXiv:2410.18745*.

Shengnan An, Zexiong Ma, Zeqi Lin, Nanning Zheng, Jian-Guang Lou, and Weizhu Chen. 2024b. Make your LLM fully utilize the context. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Dzmitry Bahdanau. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3119–3137, Bangkok, Thailand. Association for Computational Linguistics.

Avi Caciularu, Ido Dagan, Jacob Goldberger, and Arman Cohan. 2022. Long context question answering via supervised contrastive learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2872–2879, Seattle, United States. Association for Computational Linguistics.

Shijie Chen, Bernal Jiménez Gutiérrez, and Yu Su. 2024. Attention in large language models yields efficient zero-shot re-rankers. *arXiv preprint arXiv:2410.02642*.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A Smith, and Matt Gardner. 2021. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4599–4610.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Javier Ferrando, Gabriele Sarti, Arianna Bisazza, and Marta R Costa-jussà. 2024. A primer on the inner workings of transformer-based language models. *arXiv preprint arXiv:2405.00208*.

Tianyu Gao, Alexander Wettig, Howard Yen, and Danqi Chen. 2024. How to train long-context language models (effectively). *arXiv preprint arXiv:2410.02660*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910.

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

T Guo, X Chen, Y Wang, R Chang, S Pei, NV Chawla, O Wiest, and X Zhang. 2024. Large language model based multi-agents: A survey of progress and challenges. In *33rd International Joint Conference on Artificial Intelligence (IJCAI 2024)*. IJCAI; Cornell arxiv.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE.

Jinghan He, Haiyun Guo, Kuan Zhu, Zihan Zhao, Ming Tang, and Jinqiao Wang. 2024a. Seekr: Selective attention-guided knowledge retention for continual learning of large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3254–3266.

Junqing He, Kunhao Pan, Xiaoqun Dong, Zhuoyang Song, LiuYiBo LiuYiBo, Qianguosun Qianguosun, Yuxin Liang, Hao Wang, Enming Zhang, and Jiaxing Zhang. 2024b. Never lost in the middle: Mastering long-context question answering with position-agnostic decompositional training. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13628–13642, Bangkok, Thailand. Association for Computational Linguistics.

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.

Xiangyu Hong, Che Jiang, Biqing Qi, Fandong Meng, Mo Yu, Bowen Zhou, and Jie Zhou. 2024. On the token distance modeling ability of higher rope attention dimension. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 5877–5888.

Nihal Jain, Dejiao Zhang, Wasi Uddin Ahmad, Zijian Wang, Feng Nan, Xiaopeng Li, Ming Tan, Ramesh Nallapati, Baishakhi Ray, Parminder Bhatia, Xiaofei Ma, and Bing Xiang. 2023. ContraCLM: Contrastive learning for causal language model. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6436–6459, Toronto, Canada. Association for Computational Linguistics.

Ziyan Jiang, Xueguang Ma, and Wenhu Chen. 2024. Longrag: Enhancing retrieval-augmented generation with long-context llms. *arXiv preprint arXiv:2406.15319*.

Bowen Jin, Jinsung Yoon, Jiawei Han, and Sercan O Arik. 2024. Long-context llms meet rag: Overcoming challenges for long inputs in rag. *arXiv preprint arXiv:2410.05983*.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024a. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Yanming Liu, Xinyue Peng, Jiannan Cao, Shi Bo, Yanxin Shen, Xuhong Zhang, Sheng Cheng, Xun Wang, Jianwei Yin, and Tianyu Du. 2024b. Bridging context gaps: Leveraging coreference resolution for long contextual understanding. *arXiv preprint arXiv:2410.01671*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Yi Lu, Xin Zhou, Wei He, Jun Zhao, Tao Ji, Tao Gui, Qi Zhang, and Xuanjing Huang. 2024. Longheads: Multi-head attention is secretly a long context processor. *arXiv preprint arXiv:2402.10685*.

Meta. 2024. Llama 3.1 model card.

OpenAI. 2022. Introducing chatgpt.

OpenAI. 2024. Gpt-4o system card.

Alexander Peysakhovich and Adam Lerer. 2023. Attention sorting combats recency bias in long context language models. *arXiv preprint arXiv:2310.01427*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3505–3506.

Joshua David Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. 2021. Contrastive learning with hard negative samples. In *International Conference on Learning Representations*.

Sofia Serrano and Noah A. Smith. 2019. Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951, Florence, Italy. Association for Computational Linguistics.

Uri Shaham, Maor Ivgi, Avia Efrat, Jonathan Berant, and Omer Levy. 2023. ZeroSCROLLS: A zero-shot benchmark for long text understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7977–7989, Singapore. Association for Computational Linguistics.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, pages 31210–31227. PMLR.

Yixuan Su, Tian Lan, Yan Wang, Dani Yogatama, Lingpeng Kong, and Nigel Collier. 2022. A contrastive framework for neural text generation. In *Advances in Neural Information Processing Systems*.

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

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Siye Wu, Jian Xie, Jiangjie Chen, Tinghui Zhu, Kai Zhang, and Yanghua Xiao. 2024a. How easily do irrelevant inputs skew the responses of large language models? In *First Conference on Language Modeling*.

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

Wenhao Wu, Yizhong Wang, Guangxuan Xiao, Hao Peng, and Yao Fu. 2024c. Retrieval head mechanistically explains long-context factuality. *arXiv preprint arXiv:2404.15574*.

Zijun Wu, Bingyuan Liu, Ran Yan, Lei Chen, and Thomas Delteil. 2024d. Reducing distraction in long-context language models by focused learning. *arXiv preprint arXiv:2411.05928*.

Guangxuan Xiao, Jiaming Tang, Jingwei Zuo, Junxian Guo, Shang Yang, Haotian Tang, Yao Fu, and Song Han. 2024. Duoattention: Efficient long-context llm inference with retrieval and streaming heads. *arXiv preprint arXiv:2410.10819*.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *The Twelfth International Conference on Learning Representations*.

Zheyang Xiong, Vasilis Papageorgiou, Kangwook Lee, and Dimitris Papailiopoulos. 2024. From artificial needles to real haystacks: Improving retrieval capabilities in llms by finetuning on synthetic data. *arXiv preprint arXiv:2406.19292*.

Peng Xu, Wei Ping, Xianchao Wu, Chejian Xu, Zihan Liu, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Chatqa 2: Bridging the gap to proprietary llms in long context and rag capabilities. *arXiv preprint arXiv:2407.14482*.

Tianyi Lorena Yan, Fei Wang, James Y Huang, Wenxuan Zhou, Fan Yin, Aram Galstyan, Wenpeng Yin, and Muhao Chen. 2024. Contrastive instruction tuning. *arXiv preprint arXiv:2402.11138*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William 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*, pages 2369–2380.

Tianzhu Ye, Li Dong, Yuqing Xia, Yutao Sun, Yi Zhu, Gao Huang, and Furu Wei. 2024. Differential transformer. *arXiv preprint arXiv:2410.05258*.

Zifan Zheng, Yezhaohui Wang, Yuxin Huang, Shichao Song, Mingchuan Yang, Bo Tang, Feiyu Xiong, and Zhiyu Li. 2024. Attention heads of large language models: A survey. *arXiv preprint arXiv:2409.03752*.

A Retrieval Head Detection

A.1 LongBench Annotation

Each question in the HotpotQA subset of LongBench has at least two relevant passages that contain essential information to answer the question. We manually reviewed each question and annotated its golden passages. Figure 8 shows the annotation interface we used.

A.2 Retrieval Score Calculation

Assume that for a given question q, we have:

- A set of *golden* (i.e., relevant) passages, P_G.
- A set of *irrelevant* passages, P_I.

These passages are concatenated (in random order) to form the input context $\mathcal{C}$. An attention head h assigns an attention weight a_p to every passage $p \in \mathcal{C}$ (by summing the attention weights over tokens in the passage from the last token of the prompt). In the following we describe two retrieval metrics computed for head h.

EM Retrieval Score. For the EM metric, the ranking of passages by their attention weights is used directly without considering the threshold ϵ. Let

$$X = |P_G|$$

be the number of golden passages. Define a permutation σ_h that sorts the passages in $\mathcal{C}$ in descending order of attention weight:

$$a_{\sigma_h(1)} \geq a_{\sigma_h(2)} \geq \cdots \geq a_{\sigma_h(|\mathcal{C}|)}.$$

Then, for a given question q, the EM Retrieval Score for head h is defined as:

$$\text{EM}_h(q) = \begin{cases} 1, & \text{if } \{\sigma_h(1), \sigma_h(2), \ldots, \sigma_h(X)\} = P_G, \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

That is, if the top X passages (i.e., the X passages with the highest attention weights) are exactly the golden passages, we consider the retrieval perfect and set $\text{EM}_h(q) = 1$. Otherwise, $\text{EM}_h(q) = 0$. The overall EM Retrieval Score for attention head h is then the average over all test queries:

$$\mathcal{R}_h^{\text{EM}} = \frac{1}{|Q|} \sum_{q \in Q} \text{EM}_h(q), \tag{10}$$

where Q is the set of test questions.

F1 Retrieval Score. For the F1 metric, we first use a fixed threshold ϵ to decide whether a passage is *attended*. Specifically, define the set of passages attended by head h for question q as:

$$P_{A_h}(q) = \{\, p \in \mathcal{C} \mid a_p > \epsilon \,\}. \tag{11}$$

Then, we compute the precision and recall based on the golden set P_G and the attended set $P_{A_h}(q)$:

$$\text{Precision} = \frac{|P_G \cap P_{A_h}(q)|}{|P_{A_h}(q)|}, \tag{12}$$

$$\text{Recall} = \frac{|P_G \cap P_{A_h}(q)|}{|P_G|}. \tag{13}$$

The F1 Retrieval Score for head h on question q is then defined as the harmonic mean of precision and recall:

$$F1(q) = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{14}$$

with the convention that if both precision and recall are zero, we set $F1(q) = 0$. Finally, the overall F1 Retrieval Score for attention head h is obtained by averaging over all test queries:

$$\mathcal{R}_h^{\text{F1}} = \frac{1}{|Q|} \sum_{q \in Q} F1(q). \tag{15}$$

Averaging these scores over the test set Q yields $\mathcal{R}_h^{\text{EM}}$ [Eq. (10)] and $\mathcal{R}_h^{\text{F1}}$ [Eq. (15)], which serve as our final metrics for evaluating the retrieval capability of attention head h.

B More Implementation Details

We fine-tuned the Llama3.1-8B model with full parameters fine-tuning on 32 (2*16) 64G AMD INSTINCT MI250X GPUs. The distributed training was run with the DeepSpeed (Rasley et al., 2020) framework with ZeRO stage 2. The learning rate is 5e-6. We use AdamW (Loshchilov and Hutter, 2019) as the optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The template for the training sequence is: "Based on the following passages, answer the question.\n\n<passages>\n{body}\n\n</passages>\n\n Question: {question}\nAnswer: {answer}". The default EOS token is '</s>'.

C Analysis on Whole-layer Optimization

We compared the retrieval scores of attention heads by layer between the *Vanilla-SFT* baseline, *MuDAF-strong* and *MuDAF-weak*. Considering the different layers of selected attention heads, we can analyze the impact of a certain head on its layer. Table 3 shows the layer distribution of the two selection strategies. Considering some representative layers: 15, 16, 17, 25. For the layer 15, neither *MuDAF-strong* nor *MuDAF-weak* has heads from this layer, so most attention heads in this layer are not influenced with some of them being harmed actually (the line chart is below the zero threshold); For layer 16, both *MuDAF-strong* and *MuDAF-weak* select heads from it (four heads fror *MuDAF-strong* and one head for *MuDAF-weak*). As we can see, most attention heads are enhanced through *MuDAF-strong*, even though they are not selected directly, and the improvement is much bigger than *MuDAF-weak*. Meanwhile, nearly half of the attention heads also get enhanced through *MuDAF-weak*, indicating that it is also helpful by selecting just one head; Finally, for layer 17 and 25, we can clearly observe that the retrieval scores of all attention heads are significantly improved when the corresponding strategy optimizes more attention heads at that layer (i.e., *MuDAF-strong* for layer 17 and *MuDAF-weak* for layer 25). In the contrast, basically no improvement can be seen if no attention head is selected in that layer. We speculate that this phenomenon is related to the Grouped-Query Attention (GQA) (Ainslie et al., 2023), where *query heads* are divided into several subgroups and each subgroup has only one corresponding key head. It also partly explains why *MuDAF-weak* can still achieve appreciable improvements in the overall performance, given that there may exist some relatively strong attention heads within the same group.

Strategy	Layer Distribution
MuDAF-strong	Layer 13: 2 Heads Layer 16: 3 Heads Layer 17: 3 Heads
MuDAF-weak	Layer 16: 1 Head Layer 19: 1 Head Layer 25: 4 Heads Layer 27: 1 Head Layer 31: 1 Head

Table 3: The distribution of selected attention heads in two different selection strategies.

Figure 8: Our annotation interface with attention visualization for reference.

Figure 9: Retrieval scores of all attention heads listed by layer. The bars present the value of retrieval scores, while the line charts mean the different between *MuDAF-** and the *Vanilla-SFT* baseline.

D More Experimental Results

We provides full results for the ablation study on the number of selected attention heads in Table 4.

E Case Study

Figure 10 and Figure 11 show two cases that compare the output and attention distribution between Llama3.1-8B-*Vanilla-SFT* and Llama3.1-8B-*MuDAF*. *MuDAF* effectively optimizes the attention distribution of the selected attention heads, making the model be more focused on relevant passages.

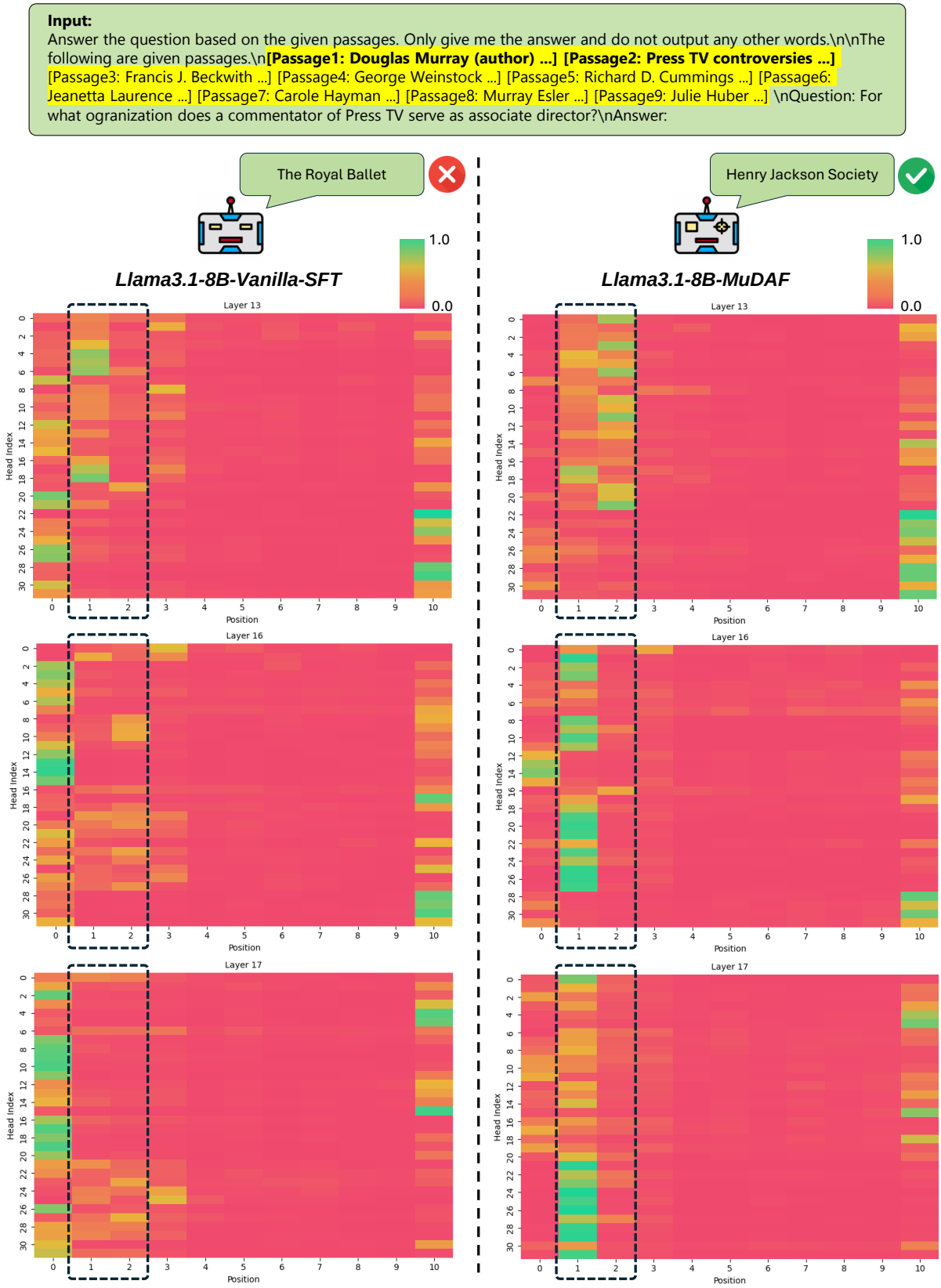

Figure 10: Comparison of the output and passage-level attention distribution (heatmaps below) in three different layers. This case contains 9 passages. The golden passages are passage#1 and passage#2 (in the dotted box). *Llama3.1-8B-MuDAF* is more focused and can redirect the attention from the beginning part (i.e., #0) to the passages.

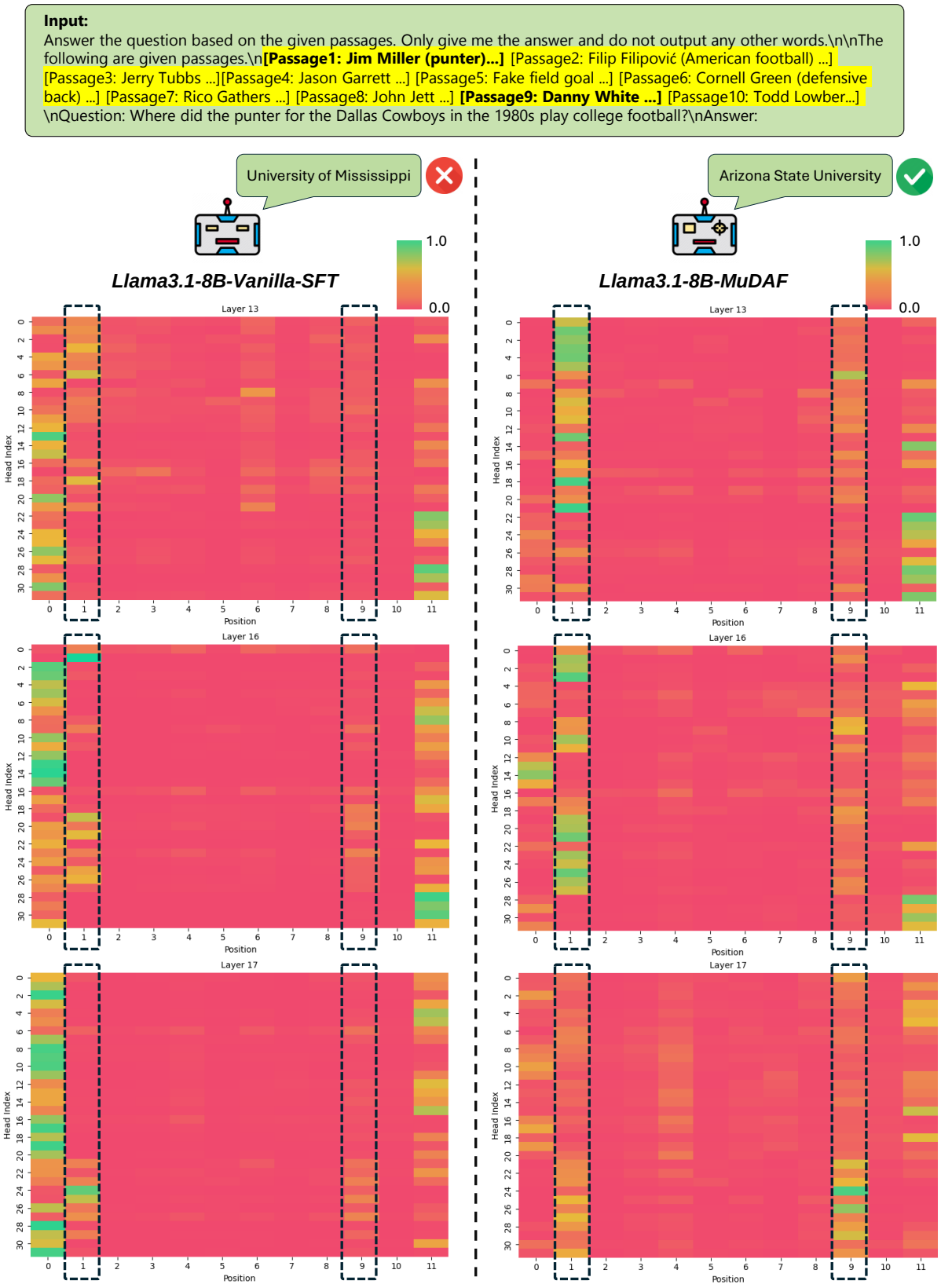

Figure 11: This case contains 10 passages. The golden passages are passage#1 and passage#9 (in the dotted box).

Num Heads	LongBench				ZeroSCROLLS		Avg.
	HotpotQA	2WikiMultihopQA	MuSiQue	Qasper	MuSiQue	Qasper	
n=0	46.8	50.5	28.9	29.2	30.1	41.6	37.8
n=2	56.8	46.5	37.8	32.9	26.8	42.6	40.5
n=4	64.7	57.9	41.7	**36.9**	23.9	43.4	44.7
n=8	71.1	64.7	47.4	36.0	35.2	42.4	49.4
n=16	72.2	**68.8**	**48.9**	33.1	28.3	40.2	48.6
n=32	**72.6**	66.1	48.3	33.2	**36.7**	40.8	49.6
n=64	69.4	65.8	48.7	**36.9**	33.8	**46.0**	**50.0**

Table 4: Ablations on the number of selected attention heads.