Enhancing Capabilities of Llama in Long Context with Dynamic Drop Attention

Anonymous ACL submission

Abstract

Large Language Models (LLMs) exhibit constrained extrapolation capabilities, particularly when confronted with input text that exceeds the model training window. This phenomenon manifests as a discernible degradation in performance, attributable to two principal factors. Firstly, the modification in positional encoding, induced by variations in text length, exerts a discernible impact on attention calculations, thereby giving rise to substantive deviations. Secondly, inherent limitations within the attention mechanism engender attention dispersion as the length of the input text increases.

In this paper, we investigate the phenomenon of attention dispersion and propose a straightforward yet effective approach, namely Dynamic Drop Attention (DDA). DDA filters noise and retains important information to mitigate attention dispersion during attention computation. DDA significantly enhances the text generation capability of LLMs without fine-tuning. To evaluate the effectiveness of the DDA, we implement it on the open-source Llama2 model and perform experiments on the LongQA and QMSum datasets. Compared to the vanilla Llama2, the DDA-based model achieves an improvement in perplexity for language modeling. Additionally, manual evaluations attest to improvements in the conciseness, relevance, and accuracy of the generated text.

1 Introduction

With the advancement of NLP technology, the capabilities of Large Language Models (LLMs) (Brown et al., 2020; Zhang et al., 2022; Touvron et al., 2023a; Ouyang et al., 2022) have become increasingly powerful, achieving astonishing performance in various NLP tasks such as question answering (Kamalloo et al., 2023), text summarization (Zhang et al., 2023; Goyal and Durrett, 2020), dialogue systems (OpenAI, 2023; Taori et al., 2023; Chiang et al., 2023), and code completion (Chen et al., 2021; Roziere et al., 2023). As LLMs are applied in real-world scenarios, the text length that LLMs need to handle in specific scenarios is also becoming longer. This requires LLMs to have effective long-text processing capabilities. For example, a research paper has about 10,000 tokens and it is a challenging task to understand the paper and generate high-quality responses to questions for Llama-2 (Touvron et al., 2023b), which is trained on a context window of 4K tokens.

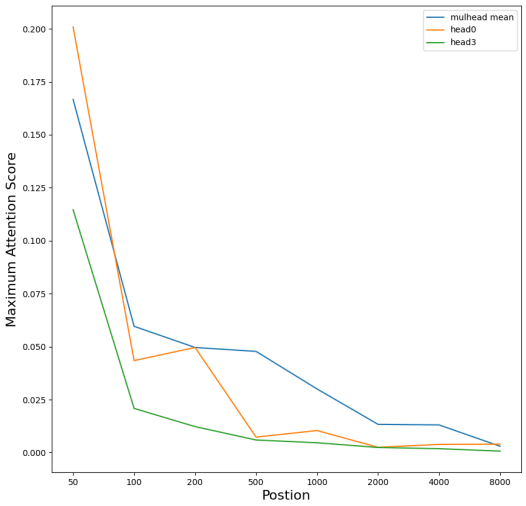
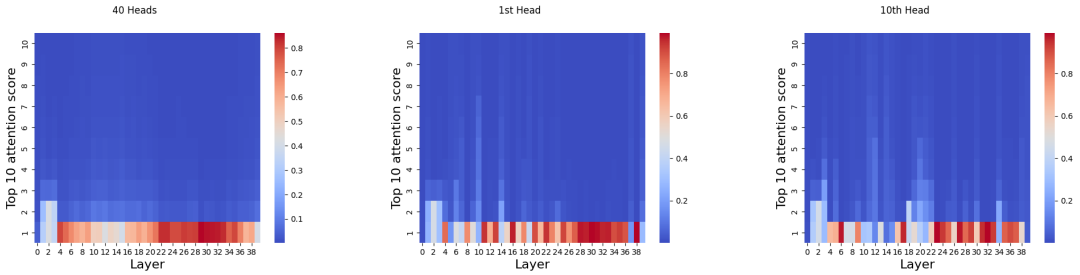


Figure 1: Depiction of Maximum Attention Scores in Relation to Text Length: A notable decline in scores is observed as the sequence length augments.

Noticeable performance degradation has been observed for LLMs when input text surpasses the model training window (Press et al., 2021; Chen et al., 2023a), attributed to two primary factors. On one hand, the capability of LLMs for long texts is affected by the position encoding which varies with length (Chen et al., 2023a), which influences the attention computation and leads to substantial deviations in attention estimation. On the other hand, the attention mechanism has inherent limita-



(a) 40 heads (b) 1st head (c) 10th head

Figure 2: Comparative visualization of the top-10 average attention logits across different attention layers for 32 sentences. Each figure corresponds to the token at the same position 8000. Fig (a) depicts the average attention score derived from multiple heads, while the remaining figures represent attention scores from individual heads. The attention scores are noticeably more evenly distributed in the bottom layers.

tions. As the input text lengthens, the phenomenon of attention dispersion(Zhao et al., 2019) arises, which leads to a decline in the model's perception of vital tokens, thereby reducing the model's ability to handle long texts.

Current works study the impact of the model's position encoding (Su et al., 2023; Chen et al., 2023a; bloc97, 2023; Xiong et al., 2023) on extrapolation to improve the capabilities of LLMs in long texts. However, the work on improving the long-text capabilities of LLMs through the attention mechanism is still insufficient. StreamingLLM (Xiao et al., 2023) analyzes the attention distribution and enhances the model's ability to generate infinitely long text based on the discovery of the attention sink phenomenon. Nevertheless, it is noteworthy that this method incurs the loss of information from preceding text segments. In this paper, we propose to explore the impact of attention dispersion on Llama-2's extrapolation ability and improve Llama-2's long text generation ability with Dynamic Drop Attention(DDA).

To explicate the attention dispersion, we calculate the average multi-head attention scores for tokens positioned between the 100th and 8000th positions across 32 sentences. As illustrated in Figure 1, it can be found that an extension in text length leads to a significant decline in average multi-head attention scores. This demonstrates that Llama-2 struggles to focus on important tokens during the processing of long texts, leading to the issue of attention dispersion.

To gain a deeper insight into the attention dispersion, we visualize the top-10 average multi-head attention scores across different layers, as illustrated in Figure 2. Each figure corresponds to the token at the identical position of 8000. Figure 2(a) shows the average attention score derived from multiple heads, while the subsequent Figure 2(b) and Figure 2(c) represent attention scores from individual heads. The attention scores are noticeably closer in numerical values in the lower layers, while they become more polarized in the higher layers. This suggests that the attention dispersion phenomenon is not prevalent across all layers. This observation arises because the initial layer exhibits minimal disparity between the embedding of the query and key, resulting in closely calculated scores. Based on the visualization, it can be tentatively concluded that the degree of attention dispersion amplifies with the increase in text length, and not all attention layers exhibit dispersion of attention.

Based on the above insights, we propose Dynamic Drop Attention(DDA), a simple yet effective method, to improve the long text generation ability of Llama-2 without fine-tuning. DDA aims to alleviate the model's attention dispersion by preserving important information and removing noise. The importance of tokens is relevant to similarity scores between query and key. To validate the effectiveness of the DDA method, experiments are performed on the LongQA (Chen et al., 2023b) and QMSum (Zhong et al., 2021) datasets. Compared to the vanilla Llama2-13B, the DDA-based Llama2 achieves a significant improvement in perplexity for generated text. Through manual evaluation, improvements in the conciseness, relevance, and accuracy of generated text are also evident.

2 Related Work

In recent years, large language models (LLMs) have achieved excellent performance in many natu-

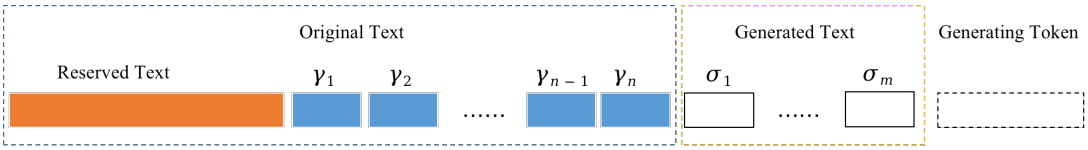


Figure 3: Overview of DDA: The yellow text represents retained content with a length that does not exceed the model training window size, requiring no further action. On the other hand, the blue original text exceeds the model training window size. It is initially segmented into n chunks, each containing L tokens, and subjected to Drop Attention (DA) with distinct drop rates. This procedure is independently applied to each chunk of the generated text.

ral language processing tasks. However, they still suffer performance limitations when dealing with long texts (Huang et al., 2023; Dong et al., 2023). This is because a fixed context length is set during model pre-training. When input texts far exceed this length appear in downstream tasks, the model cannot extrapolate to untrained position encoding, leading to performance degradation. Recently, many works have been proposed for handling long texts, which can be mainly divided into three lines: *segmentation*, *position encoding extrapolation*, and *attention mechanism improvement*.

Segmentation Segmentation-based methods divide long text into multiple segments equal to the model's context window and enable information flow between different segments through certain mechanisms. Transformer-XL (Dai et al., 2019) adopts a segment recursion mechanism, reusing the hidden state of previous segments when modeling the current segment. RMT (Bulatov et al., 2022) adds memory tokens at the beginning and end of the segment, where the read memory token at the beginning of the segment can read information from the previous segment, and the write memory token at the end is used to update the memory representation. AutoCompressors (Chevalier et al., 2023) add a summary token at the end of each segment and accumulate it as a soft token in all subsequent segments. LongLoRA (Chen et al., 2023b) adopts a shift-short attention mechanism, it achieves token overlap between two segments through shift operations, which allows attention calculations to be performed between segments. Although these methods can integrate information from all segments to some extent, the segmentation of continuous content inevitably leads to information loss. On the other hand, to implement special segment information transmission mechanisms, it is necessary to modify the model structure and fine-tune the model with training data, which is a time-consuming process.

position encoding extrapolation The goal of these methods is to extrapolate the finite position encoding trained by the model to infinite length, thereby enabling the model trained based on a shorter context window to handle longer texts. Rotary Position Embeddings (RoPE) (Su et al., 2023) use a rotation matrix to integrate relative position information dependence into the calculation of self-attention, which is a relative position encoding method with good extrapolation capabilities. NTK-aware RoPE is a further extension of RoPE. It uses a set of trigonometric function vectors with different periods to express positions, and this method does not require fine-tuning of the model. Positional Interpolation (PI) (Chen et al., 2023a) compresses position encoding uniformly into the range of trained position encoding, but it requires a small amount of data fine-tuning to achieve decent results.

Attention mechanism improvement These methods focus on enhancing the model's ability to capture richer information within the context window. It has an orthogonal relationship with the methods mentioned above and can be easily combined with the methods of the first two lines. Streaming-LLM (Xiao et al., 2023) points out that autoregressive LLMs exhibit the phenomenon of "attention sink", namely attention scores are concentrated on the initial tokens. This is determined by the characteristics of autoregressive language modeling because the initial tokens can see all subsequent tokens. This leads to a sharp drop in the performance of methods based on sliding window attention after the key value of the initial token is missing. Based on this discovery, Streaming-LLM retains the key value of the initial tokens for subsequent window attention calculations of the sliding window attention mechanism, effectively enhancing the model's ability to capture long-text information. Our method belongs to this line. Similar to streaming-LLM, we found that LLMs exhibit attention dispersion when dealing with long texts, and based on this discovery, we proposed the drop

attention method to filter low-relevance tokens to alleviate the attention dispersion issue.

3 Method

3.1 Attention dispersion

As the length of the input text extends, a phenomenon known as attention dispersion emerges, leading to a decrease in the model's ability to perceive crucial tokens. As shown in Figure 1, the degree of attention dispersion intensifies in correlation with the increase in text length. A significant degradation in performance becomes apparent when the input text exceeds the model's training window. The attention dispersion is tied to the attention computation mechanism. To explore why the degree of attention dispersion increases as the text length increases, an analysis of the attention computation mechanism (Vaswani et al., 2023) is conducted as follows:

$$attn(q,k,v) = softmax(\frac{q*k}{\sqrt{d}})v \quad (1)$$

The definition of softmax is as follows:

$$a_i = \frac{e_i}{e_i + \sum_{j=0,j\neq i}^{n} e_j} \quad (2)$$

In practical NLP tasks, the text length n is not fixed. When dealing with some long texts, as the text length n gradually increases, there are two trends in the maximum value a_i of the attention coefficient, as shown below:

$$\begin{cases} a_i \to 1, e_i \gg e_j \\ a_i \to 1/n, e_i \simeq e_j \end{cases} \quad (3)$$

As shown in Figure 2, the first scenario mostly occurs in the lower-level attention, while the second scenario is more prevalent in the upper-level attention. As n increases, the a_i in the first case is less affected; although it changes, it remains a relatively large constant value close to 1. On the other hand, the a_i in the second case will be significantly affected, with a_i decreasing significantly, even approaching 0, as n tends towards infinity. Specifically, as illustrated in Figure 1, when the text length n reaches 8000, the maximum value of the attention coefficient is 0.003, representing a two-order-of-magnitude drop compared to the maximum value of the attention coefficient at position 50. Intuitively, the decrease in the value of a_i will introduce more noise into the attention mechanism when aggregating vector information, reducing the model's resolution and thereby affecting the inference performance.

3.2 Drop Attention

To alleviate model attention dispersion, an intuitive idea is to reduce the number of tokens calculating attention scores. To remove noise and retain more important information, we propose a simple and effective method called Drop Attention (DA). The specific operation is as follows: first, calculate the attention scores $a = [a_0, a_1, ..., a_l]$ between query and key, then calculate the attention coefficient a_q corresponding to the quantile q of this vector, and call q the drop rate. The attention coefficients smaller than a_q are set to $-\infty$, which can be represented by the following equation:

$$a_i = \begin{cases} a_i, \ if \ a_i > a_q \\ -\infty, \ if \ a_i <= a_q \end{cases} \quad (4)$$

By utilizing this approach, we can effectively filter out noise and retain crucial information. Furthermore, it allows for the mitigation of attention dispersion, thereby enhancing the long-text question-answering proficiency of Llama2.

3.3 Dynamic Drop Attention

Equation 2 suggests that as the sentence length increases and the attention coefficient diminishes, the degree of attention dispersion escalates. There is a positive correlation between the degree of attention dispersion and length, implying that longer sentences exhibit more pronounced attention dispersion. Consequently, we introduce dynamic drop attention (DDA), a method where different positions are assigned varying drop rates during the calculation of attention.

As shown in Figure 3, the yellow part of the original text represents retained content with a length that does not exceed the model training window size, the part of the original text has a normal calculation of attention scores. The blue chunks of original text exceed the model training window size. The text is divided into different chunks, each chunk may contain L tokens. When calculating attention scores, each chunk needs DA with a distinct drop rate. Drop rate γ_i is set as follow:

$$\gamma_i = \gamma_{i-1} + \epsilon, i = 0, 1....n \quad (5)$$

A relatively small initial drop rate γ_1 is set in the first chunk. As the length of the tokens that need to

be dropped increases, the drop rate of subsequent chunks adds a small constant ϵ to the initial drop rate, ultimately forming a segmented dynamic drop rate. For the generated text, it is also divided into different chunks and uses different drop rates σ_i.

4 Experiments

To evaluate the effectiveness of DA and DDA, we apply these methods to Llama2-13b-chat and compare them with the vanilla Llama2-13b-chat. The evaluation involves using the LongQA dataset (Chen et al., 2023b) and the QMSum dataset (Zhong et al., 2021). LongQA comprises over 7,000 question-answer pairs, with questions generated from books or articles to test the models. The average length of the text in LongQA is 10,500 tokens. QMSum consists of 1,808 query-summary pairs from 232 meetings across multiple domains, with an average meeting length of 9,070 tokens. In the experiment, due to limitations in GPU memory, we filter out examples with a token count exceeding 12,000 and simultaneously excluded samples with a length less than 4,000 tokens. All experiments are carried out using 8 GPUs of A100*80G. No fine-tuning is applied in any of the experiments.

4.1 Comparison with vanilla Llamm2-13B

To evaluate the efficacy of DA and DDA, we conduct assessments on the LongQA and QMSum datasets, utilizing log PPL as the evaluation metric. Both methods are applied to the last 4000 tokens of the original text. For DA, a fixed drop rate of 0.3 is employed, while DDA utilizes an initial drop rate of 0.05, incrementing by 0.05 for every 1000 tokens. As presented in Table 1, both

Table 1: Comparison of log PPL scores among Llama2-13B, Llama2-DA, and Llama2-DDA. A significant decrease in the log PPL score is observed for Llama2-DDA.

metrics	Llama2-13B	Llama2-DA	Llama2-DDA
LongQA	1.691	1.681	**1.668**
QMSum	2.588	2.576	**2.572**

DA and DDA result in a reduction in log PPL compared to the baseline Llama2-13B model. This reduction suggests that alleviating attention dispersion is crucial for enhancing the model's ability to handle long texts. In the LongQA dataset, Llama2-DDA achieves a log PPL score of 1.668, significantly lower than both Llama2-13B (1.691) and Llama2-DA (1.681). Similarly, in the QMSum dataset, Llama2-DDA exhibits a log PPL score of 2.572, showing a significant decrease compared to both Llama2-13B (2.588) and Llama2-DA (2.576). The result underscores the effectiveness of DDA in reducing attention dispersion, contributing to enhanced model understanding and performance on LongQA and QMSum datasets.

Furthermore, the DDA-based Llama demonstrates a lower PPL score compared to the DA-based Llama, suggesting that different positions experience varying degrees of attention dispersion, necessitating distinct drop rates.

To further evaluate the generation performance, we randomly select 100 samples from the generated results for manual evaluation. To ensure fairness, we anonymize the generated results before having them assessed by two researchers. The evaluation focuses on the relevance of the generated content to the questions and the presence of redundancy. The evaluation results are shown in Figure 4. In the set of 100 generated samples, the majority exhibit a similar text generation performance between the two models. Llama2-13B-DA shows a slight improvement compared to Llama2-13B, while Llama2-13B-DDA demonstrates a more significant improvement over the vanilla Llama2-13B. The manual evaluation results also indicate a noticeable enhancement in the long-text processing capability of Llama2-13B with DDA.

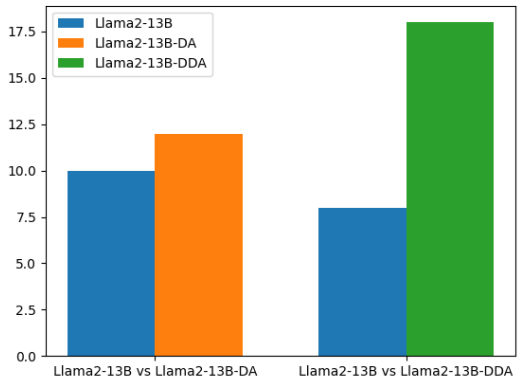


Figure 4: Human evaluation results of 100 generated texts, excluding draws. The x-axis denotes the number of victories in human evaluation. A significant improvement is observed in Llama2-13B with DDA compared to the vanilla Llama2-13B.

Original Text			Generated Text	PPL
initial drop rate	increment	max drop rate	drop rate	
0	0	0	0	1.691
0	0	0	0.1	1.694
0	0	0	0.3	1699
0	0	0	0.5	1.720
0.3	0.0	0.3	0	1.674
0.1	0.05	0.25	0	1.669
0.1	0.1	0.4	0	1.670
0.15	0.05	0.3	0	**1.668**
0.15	0.05	0.3	0.3	1.684

Table 2: Effect of drop rate on PPL in Llama2-13B Model

4.2 Different Number of Tokens with DA

An examination of the impact of DA on varying token counts is underway to evaluate its effectiveness. Specifically, we implement DA on the last 2000, 4000, and 6000 tokens of the original texts sourced from the LongQA dataset. The corresponding log PPL scores for each token count are summarized in Table 3. Notably, the model attains its lowest PPL score of 1.68 when the number of tokens is set to 6000. Additionally, the PPL of the last 4000 tokens using DA is in proximity to that of the last 6000 tokens using DA.

This observation can be elucidated by considering that the average length of the dataset's text is 10500, while the training window of Llama2-13B is constrained to 4000 tokens. The last 6000 tokens may not have undergone sufficient training, potentially leading to attention dispersion. The application of DA to these tokens appears to effectively enhance the generation performance, as indicated by the observed decrease in the log PPL score.

Table 3: PPL scores corresponding to different number of tokens utilizing DA.

tokens	0	2000	4000	6000
log PPL	1.691	1.684	1.681	1.68

4.3 Select Proper Drop Rate

The careful selection of an appropriate drop rate is crucial for DA and DDA. An inappropriate drop rate may result in a performance degradation in the Llama2-13B model. To investigate the impact of different drop rates, a series of experiments are conducted, utilizing PPL as the evaluation metric. The results are summarized in Table 2.

As evident from the table, the configuration with an initial drop rate of 0.15, an increment of 0.05, and a maximum drop rate of 0.3 yields the lowest perplexity value for the generated text, recording a value of 1.668. This particular setting outperforms all other parameter combinations, indicating its effectiveness in optimizing text generation quality. Additionally, the implementation results also reveal that the effectiveness of DDA surpasses that of DA. Choosing an appropriate drop rate is crucial, as both excessively large and excessively small drop rates may fail to achieve the optimal improvement. Currently, a drop rate of 0.3 appears to be relatively effective.

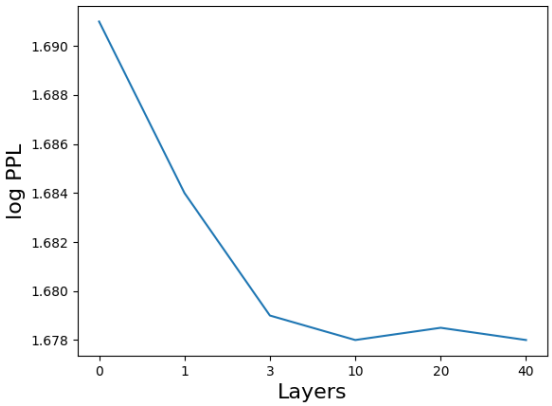


Figure 5: PPL of different layers with DA

Surprisingly, the counterintuitive observation emerges that DA applied on the generated text leads to an increase in PPL compared to the baseline Llama2-13B model. Interestingly, even when both DDA and DA are applied to the original and generated texts, the PPL score increases compared to the scenario where DDA is exclusively applied to the original text. This implies a significant influence on

Table 4: The maximum attention score of Llama2 and Llama2-DDA.The results validate our hypothesis that DDA can effectively alleviate the phenomenon of attention dispersion

method	3000	2500	2000	1500	1000	500	0
Llama2	0.005703	0.002293	0.002922	0.003391	0.00422	0.002266	0.002367
Llama2-DDA	0.005848	0.00245	0.003096	0.003635	0.004414	0.002594	0.002733
Relative Change	+2.54%	+6.84%	+5.95%	+7.19%	+4.60%	+14.47%	+15.46%
Drop Rate	15%	20%	20%	25%	25%	30%	30%

the model's performance when DA is implemented on the generated text, underscoring the intricate interplay between attention mechanisms and model behavior.

4.4 The layers of Attention Dispersion

To further investigate the mechanism of attention dispersion in Llama2, we visualize the top-k attention scores of the $8000th$ token, as depicted in Figure 6. In the lower layers of attention, the maximum attention scores exhibit relatively small values, and the differences between these values are also minimal. As the attention layers deepen, the disparities between the top-k attention values increase, and Llama2's attention starts to focus on a select few tokens. This suggests that attention dispersion predominantly occurs in the lower layers of attention.

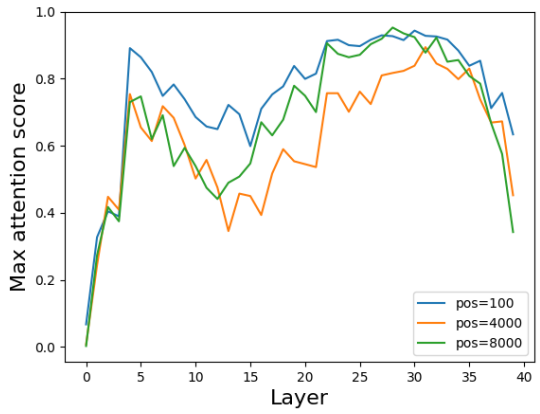


Figure 6: Average of maximum attention scores in different layer

To assess whether the lower-level attention layers are more susceptible to attention dispersion, attention layers one, three, ten, twenty, and forty underwent DA with a drop rate of 0.3. As illustrated in Figure 5, which showcases the PPL scores after applying DA to different attention layers, the lowest PPL score is attained when three attention layers undergo DA. With an increase in the number of DA layers, the PPL score remains almost unchanged. These experimental findings align with the earlier observation that attention dispersion is primarily concentrated in the lower-level attention. Therefore, by applying drop attention to the first three attention layers, the attention dispersion phenomenon in Llama2-13B is mitigated, thereby enhancing its text generation capability.

The reason why attention is comparatively more dispersed in the lower layers is that, at the initial computation stage, the disparity between the query vector and the key embedding is not significant. Consequently, during the calculation of attention scores, the lower layers of attention yield relatively mild and evenly distributed probability values. However, as the attention layers stack and the aggregation of vectors intensifies, the differences between vectors become more pronounced, leading to the polarization of the probability distribution.

4.5 Alleviating Attention Dispersion

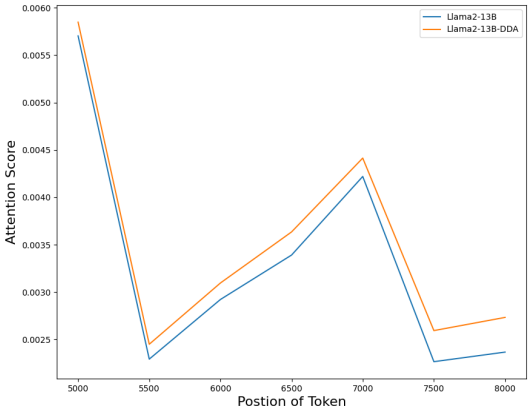


Figure 7: Comparative analysis of the maximum attention score for the same token position ranging from 5000 to 8000, between the vanilla Llama2 and Llama2 implemented with DDA. The trend suggests that DDA assists Llama2 in mitigating attention dispersion.

In order to assess the effectiveness of DDA in mitigating attention dispersion, we conducted an analysis comparing changes in attention coefficients between Llama2 and Llama2-DDA for specific token positions. The evaluation focused on the last 3000 tokens, and we computed the average of the top 5 maximum multi-head attention coefficients for every 500 tokens. The summarized results are presented in Table 4 and illustrated in Figure 7.

The Table4 and Figure7 show that the attention scores for Llama2-DDA are generally higher than those for the original Llama2, indicating a significant alleviation of attention dispersion after DDA processing. This improvement is particularly noticeable for tokens in later positions, suggesting that DDA has a more pronounced impact when addressing cases of severe attention dispersion. These findings support the hypothesis that DDA effectively mitigates attention dispersion issues, especially in contexts where dispersion is more prominent.

In summary, the results validate our hypothesis that DDA can effectively alleviate the phenomenon of attention dispersion, leading to improvements in the model's performance.

5 Conclusion

In conclusion, the current state of LLMs reveals a limitation in their ability to extrapolate text length. The performance degradation observed when input text surpasses the model training window is attributed to two key factors. Firstly, alterations in positional encoding, induced by length variations, disrupt attention calculations, leading to significant deviations in attention estimation and adversely affecting the LLMs' proficiency in handling extended texts. Secondly, inherent constraints within the attention mechanism result in attention diffusion as the input text lengthens, diminishing the model's awareness of crucial tokens and, consequently, impeding its effectiveness in processing lengthy texts.

The research delves into the phenomenon of attention dispersion and introduces an uncomplicated yet effective method DDA. DDA dynamically filters noise and retain important information to alleviate attention dispersion during attention computation in LLMs. The method notably enhances the text generation capability of LLMs without necessitating fine-tuning. To validate the efficacy of the DDA method, we apply it to the open-source Llama2-13B model and conduct experiments on the LongQA and QMSum dataset. The DDA-based Llama2 demonstrates a discernible decrease in PPL for generated text when compared to the vanilla Llama2-13B. The manual evaluation further corroborates improvements in the conciseness, relevance, and accuracy of the generated text.

6 Limitations

This article still has some limitations. Firstly, the study reveals a significant decline in the model's generation performance when DDA is applied to generated text. However, the paper does not delve into the underlying reasons for this phenomenon. Secondly, there is room for improvement in the computational efficiency of the DDA method. The current implementation is relatively slow and incurs a large GPU memory footprint, especially when handling longer texts, leading to the risk of Out-Of-Memory (OOM) issues. Lastly, considering the application of DDA during the model's finetuning and pretraining stages may be beneficial to evaluate its impact on the model's extrapolation ability when dealing with long texts. Further research and optimization in these areas would enhance the comprehensiveness and practicality of the proposed method.

References

bloc97. 2023. Ntk-aware scaled rope allows llama models to have extended (8k+) context size without any fine-tuning and minimal perplexity degradation.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Aydar Bulatov, Yuri Kuratov, and Mikhail S. Burtsev. 2022. Recurrent memory transformer.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. 2023a. Extending context window of large language models via positional interpolation. *arXiv preprint arXiv:2306.15595*.

Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. 2023b. Longlora: Efficient fine-tuning of long-context large language models.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. 2023. Adapting language models to compress contexts.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context.

Zican Dong, Tianyi Tang, Lunyi Li, and Wayne Xin Zhao. 2023. A survey on long text modeling with transformers.

Tanya Goyal and Greg Durrett. 2020. Evaluating factuality in generation with dependency-level entailment.

Yunpeng Huang, Jingwei Xu, Zixu Jiang, Junyu Lai, Zenan Li, Yuan Yao, Taolue Chen, Lijuan Yang, Zhou Xin, and Xiaoxing Ma. 2023. Advancing transformer architecture in long-context large language models: A comprehensive survey.

Ehsan Kamalloo, Nouha Dziri, Charles LA Clarke, and Davood Rafiei. 2023. Evaluating open-domain question answering in the era of large language models. *arXiv preprint arXiv:2305.06984*.

OpenAI. 2023. Gpt-4 technical report.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Ofir Press, Noah A Smith, and Mike Lewis. 2021. Train short, test long: Attention with linear biases enables input length extrapolation. *arXiv preprint arXiv:2108.12409*.

Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, et al. 2023. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2023. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, page 127063.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2023. Efficient streaming language models with attention sinks.

Wenhan Xiong, Jingyu Liu, Igor Molybog, Hejia Zhang, Prajjwal Bhargava, Rui Hou, Louis Martin, Rashi Rungta, Karthik Abinav Sankararaman, Barlas Oguz, et al. 2023. Effective long-context scaling of foundation models. *arXiv preprint arXiv:2309.16039*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B Hashimoto. 2023. Benchmarking large language models for news summarization. *arXiv preprint arXiv:2301.13848*.

Guangxiang Zhao, Junyang Lin, Zhiyuan Zhang, Xuancheng Ren, Qi Su, and Xu Sun. 2019. Explicit sparse transformer: Concentrated attention through explicit selection. *arXiv preprint arXiv:1912.11637*.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, et al. 2021. Qmsum: A new benchmark for query-based multi-domain meeting summarization. *arXiv preprint arXiv:2104.05938*.