ZigzagAttention: Efficient Long-Context Inference with Exclusive Retrieval and Streaming Heads

Anonymous ACL submission

Abstract

With the rapid development of large language models (LLMs), handling long context has become one of the vital abilities in LLMs. Such long-context ability is accompanied by difficulties in deployment, especially due to the increased consumption of KV cache. There is certain work aiming to optimize the memory footprint of KV cache, inspired by the observation that attention heads can be categorized into retrieval heads that are of great significance and streaming heads that are of less significance. Typically, identifying the streaming heads and and waiving the KV cache in the streaming heads would largely reduce the overhead without hurting the performance that much. However, since employing both retrieval and streaming heads in one layer decomposes one large round of attention computation into two small ones, it may unexpectedly bring extra latency on accessing and indexing tensors. Based on this intuition, we impose an important improvement to the identification process of retrieval and streaming heads, in which we design a criterion that enforces exclusively retrieval or streaming heads gathered in one unique layer. In this way, we further eliminate the extra latency and only incur negligible performance degradation. Our method named ZigzagAttention is competitive among considered baselines owing to reduced latency and comparable performance.

1 Introduction

In recent years, large language models (LLMs) (Dubey et al., 2024; Liu et al., 2024) have demonstrated significant potential across diverse domains (Chiang et al., 2023). However, the generation process of LLMs is inherently sequential. The sequential nature inevitably leads to substantial serving latency, particularly in scenarios involving long contexts.

The primary challenge of serving LLMs for long-context applications lies in the $O(n^2)$—where n denotes the sequence length—complexity of attention (Vaswani, 2017). The inference can be divided into two phases, i.e., prefilling phase and decoding phase. Essentially, in the decoding phase, a linear increase in memory would be natural due to the use of the key-value (KV) cache technique, which stores intermediate representations of previously seen tokens to reduce latency. In long-context scenarios, the memory of the KV cache can even exceed that of the model itself (Liu et al., 2023).

To address the memory burden imposed by KV cache, numerous approaches have been proposed to optimize the KV cache from various perspectives. Among these, DuoAttention (Xiao et al., 2024) is a typical representative. DuoAttention intends to identify *retrieval heads* (Wu et al., 2024) that are of great importance for long-context modeling and *streaming heads* (Xiao et al., 2023) that are of less importance, and predominantly waive the KV cache in the streaming heads. In doing so, DuoAttention has preserved the long-context capabilities of LLMs while improving computational efficiency.

However, DuoAttention requires processing attention computations twice separately for retrieval heads and streaming heads within one layer. Unfortunately, such separation necessitates additional memory accessing and introduces unwanted tensor indexing, leading to increased latency. This overhead becomes pronounced particularly along the expansion of context. Based on the intuition, we propose a valuable rearrangement of the retrieval and streaming heads. By enforcing either retrieval or streaming heads mutually exclusive across layers, we can perform one attention computation at each layer, thereby avoiding extra latency associated with redundant memory accessing and tensor indexing.

On an extensive set of experiments ranging from LongBench to Needle-in-a-Haystack, our proposed method ZigzagAttention achieves competitive performance while significantly reduced latency.

2 ZigZagAttention

2.1 Preliminary

To identify retrieval and streaming heads in a LLM, DuoAttention firstly plugs an importance score $\alpha \in [0, 1]$ onto each attention head, secondly employs a distillation-driven training on a synthetic dataset curated in the form of long-context passkey retrieval, and finally determines retrieval and streaming heads based on the descending order of the converged α values and a predefined quantile.

Specifically, α for each head is initialized to 1, and constrained to the range $[0, 1]$. During the training, it performs attention computation twice in each forward pass: one using full attention (corresponding to retrieval head), and another using streaming attention (corresponding to streaming head). This is formalized as follows:

$$\begin{aligned}\mathsf{attention}_{i,j} = \quad & \alpha_{i,j} \cdot \mathsf{full\_attention} \quad + \\ & (1 - \alpha_{i,j}) \cdot \mathsf{streaming\_attention}\end{aligned} \tag{1}$$

where i and j denote the layer index and the attention head index within a layer, respectively. A synthetic dataset is used, with passkeys inserted at varying depths in the sequence, as the training task. The distillation-like training objective is formulated as follows:

$$\mathcal{L}_{\mathsf{dist}} = \frac{1}{K} \sum_{k=1}^{K} \sum_{t=T-R+1}^{T} (\mathbf{h}_{\mathsf{full}}^{(k)}[t] - \mathbf{h}_{\mathsf{mix}}^{(k)}[t])^2 \tag{2}$$

where K represents the dimension of hidden states, T denotes the total sequence length, and R means to the response length of the sequence. $\mathbf{h}_{\mathsf{full}}$ and $\mathbf{h}_{\mathsf{mix}}$ refer to the final hidden states from the standard full attention and the mixed attention computed in Equation 1, respectively. To ensure sparsity in α, an additional L_1 regularization term (Tibshirani, 1996) is added:

$$\mathcal{L}_{\mathsf{reg}} = \sum_{i=1}^{L} \sum_{j=1}^{H} |\alpha_{i,j}| \tag{3}$$

where L is the number of layers in the model, and H is the number of attention heads per layer. The final loss function is formulated as:

$$\mathcal{L}_{\mathsf{duo}} = \mathcal{L}_{\mathsf{dist}} + \lambda \mathcal{L}_{\mathsf{reg}} \tag{4}$$

where λ is a coefficient controlling the impact of the regularization term. After training, the attention heads are sorted based on their final α values. By specifying a custom sparsity quantile, the heads can be categorized as either full attention (retrieval heads) or streaming attention (streaming heads).

As we can observe in DuoAttention, if a quantile is defined to categorize attention heads, different attention heads are likely to coexist within one layer. This kind of allocation may introduce extra latency due to the need for separate computations for each type of attention head.

2.2 Transport Optimization

To alleviate the need of two rounds of attention computations, we consider the most straightforward way to achieve so. That is, leveraging the converged α values from DuoAttention, and defining the transition from DuoAttention to ZigzagAttention a transport optimization problem. Provided that the original sparsity (or say the proportion of streaming heads) in DuoAttention is s, accordingly in ZigzagAttention, the number of layers corresponding to all streaming heads should be p where $p/L = s$, and the number of layers corresponding to all retrieval heads should be $q = L - p$.

In the transport optimization problem, there is a operation set $O = o_{i,j}$ comprising of totally $L \cdot H$ operations need to be carried out, and three operations are defined: 1) maintaining the type of attention head $o^{(0)}$, 2) turning a retrieval head to streaming one $o^{(1)}$, and reversely turning a streaming head to retrieval one $o^{(2)}$. Ideally, shifting from a retrieval head to streaming head would lead to performance decline, while shifting from a streaming head would lead to performance boost. Thereby, the optimization objective is shown below:

$$\begin{gathered}\min_{o_{i,j}} \mathcal{L}_{\mathsf{zigzag}} \quad s.t. \quad p + q = L \\ \mathcal{L}_{\mathsf{zigzag}} = \sum_{i=1}^{L} \sum_{j=1}^{H} \hat{\alpha}_{i,j} \\ \hat{\alpha}_{i,j} = \begin{cases} 0, & o_{i,j} = o^{(0)} \\ \alpha_{i,j}, & o_{i,j} = o^{(1)} \\ -\omega \cdot \alpha, & o_{i,j} = o^{(2)} \end{cases}\end{gathered} \tag{5}$$

Enumeratively, the number of possible combinations under the subjection $p + q = L$ is $\binom{f}{L}$. Once one of these combinations is used, then the operation set O should also be determined. Since $\binom{f}{L}$ is computationally trackable, we empirically examine each of them one by one and uncover the one yielding the minimum. $\omega \in [0, 1]$ represents a scaling factor, which is determined through grid search to identify its optimal value.

2.3 Fine-tuning for Enhanced Ability

After optimization, we observe that while the performance is comparable to the baseline, optionally, fine-tuning with minimal training cost can further enhance performance on certain benchmarks, particularly retrieval tasks.

For fine-tuning, we adopt the previously used training scheme in DuoAttention and plug the layer-wise α_ls onto layers rather than heads using the optimal combination from the aforementioned transport optimization. By leveraging these trained results, we can sparsify the model to achieve improved performance.

3 Experiments

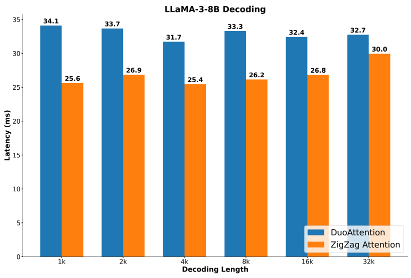

Figure 1: Per token decoding latency. The prefilling length here is set to 16k, and the decoding length varies from 1k to 32k.

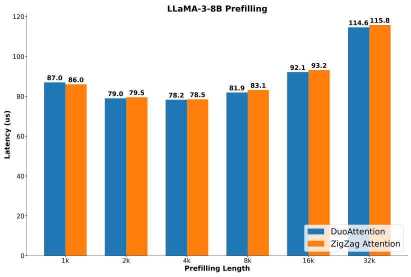

Figure 2: Per token prefilling latency. The decoding length here is set to 1k, and the prefilling length varies from 1k to 32k.

3.1 Settings

We conduct experiments using the long-context extension version of the LLaMA-3-8B (Dubey et al., 2024; Pekelis et al., 2024) model and evaluate its performance on both long-context and short-context benchmarks to ensure a comprehensive assessment. For long-context benchmarks, we select LongBench (Bai et al., 2023) and Needle-in-a-Haystack (Kamradt, 2024), while for short-context benchmarks, we choose MMLU (Hendrycks et al., 2020), BBH (Suzgun et al., 2022), and DROP (Dua et al., 2019). To evaluate efficiency, we test the model's performance across various combinations of prefilling and decoding lengths to minimize the impact of measurement errors. The primary training settings are aligned with those used in DuoAttention.

3.2 Efficiency Results

The results are illustrated in Figure 1, while DuoAttention demonstrates lower latency and better performance compared to the original model, ZigZagAttention achieves even lower latency across all decoding lengths. ZigZagAttention achieves up to 37% acceleration in 1k context length. Figure 2 compares the per-token prefilling latency between ZigZagAttention and DuoAttention, since ZigZagAttention do not modify the prefilling stage, our method maintains normal prefilling speed. This confirms that ZigZagAttention does not introduce additional latency during the prefilling stage. As for the time cost of the transport problem, the total time cost with our optimized method is around **7 minute**.

3.3 Long Context Benchmark

For this evaluation, we applied a 50% sparsity level for the LLaMA-3-8B model and set the sink size to 128 and window length to 256 for streaming attention.

LongBench The results for selected datasets are presented in Table 1, while average scores across all tasks are shown in Table 2. From Table 1, importantly, there is no significant decline in metrics compared to the original model, demonstrating that ZigZagAttention can effectively manage long-context situations. As shown in Table 2, ZigZagAttention scores are only marginally lower, compared to DuoAttention and the original model.

Needle-in-a-Haystack (NIAH) Figure 3 illustrates that ZigZagAttention performs exceptionally well across various context lengths ranging from 40k to 280k tokens. The results indicate that ZigZagAttention successfully discards unimportant KV caches during inference, without any

Table 1: Evaluation results on LongBench. Here "LM-3" refers to the results of LLaMA-3-8B long context extension version, "DA" refers to DuoAttention, "ZA" refers to ZigZag Attention.

Method	Single-Document QA			Multi-Document QA			Summarization			Few-shot Learning			Synthetic		Code	
	NrtvQA	Qasper	MF-en	HotpotQA	2WikiMQA	Musique	GovReport	QMSum	MultiNews	TREC	TriviaQA	SAMSum	PCount	PRe	Lcc	RB-p
LM-3	26.84	29.32	52.86	40.87	28.86	24.68	34.25	24.58	27.8	71.0	87.7	41.95	1.0	79.0	37.91	37.71
DA	25.72	28.35	49.75	43.28	29.9	23.41	32.34	24.69	28.06	72.0	86.85	41.97	1.5	83.12	38.33	39.5
ZA	22.53	23.7	49.89	38.53	23.61	21.21	30.62	24.16	27.12	71.0	82.22	40.85	1.0	85.0	45.38	44.7

Table 2: The average scores on overall LongBench.

Method	Budget	LongBench
LM-3	100%	39.78
DA	50%	39.45
ZA	50%	38.44

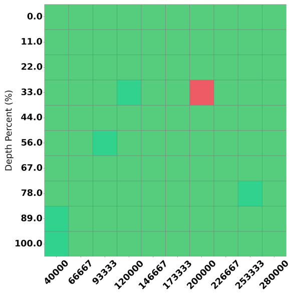

Figure 3: Results on NIAH varies from 40k to 280k.

performance degradation in complex long-context retrieval tasks.

3.4 Short Context Benchmark

Table 3: Evaluation results on MMLU, BBH and DROP benchmarks.

Method	Budget	MMLU	BBH	DROP
		5-shot	3-shot	3-shot
LM-3	100%	62.31	41.95	44.18
DA	50%	62.56	42.14	42.07
ZA	50%	62.31	42.03	43.50

As shown in Table 3, ZIGZAGATTENTION demonstrates performance comparable to that of the base model LLaMA-3 across these important benchmarks. This indicates that the ZIGZAGATTENTION mechanism does not impair the model's basic capabilities.

3.5 Ablation Study

Impact of ω Changes in ω can alter the combination of layers, potentially affecting the model's performance in long-context situations and benchmarks. Specifically, when ω is set to 0.2, 0.3, or 0.4, the final combinations are identical to those obtained with $w = 0.1$. In Table 4, it is evident that $w = 0.1$ yields the optimal combination with the best performance across multiple tasks in long-context benchmarks.

Table 4: The average scores on overall LongBench.

ω	Budget	LongBench
0.1	50%	38.44
0.5	50%	37.59
0.6	50%	37.08
0.7	50%	37.05
0.8	50%	35.27
0.9	50%	36.02

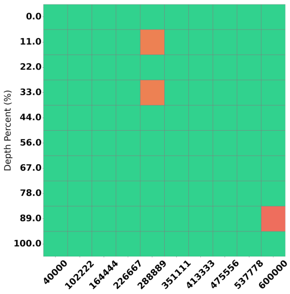

Figure 4: Results for after training in ZA. We successfully extent the context length to 600k.

Context Length Extension with Fine-tuning As shown in Figure 4, the fine-tuning allowed us to extend the context length from 280k tokens to 600k tokens with minimal additional training.

4 Conclusion

In this paper, we introduced ZIGZAGATTENTION, a method built upon DuoAttention and designed to address the challenges of handling long-context situations. Our results demonstrate that ZIGZAGATTENTION achieves performance comparable to the original model, indicating that it can significantly lower latency without degrading model capabilities.

Limitations

In this paper, we propose ZIGZAGATTENTION to accelerate model inference. However, the current method has certain limitations. In terms of efficiency, the speedup ratio decreases for longer decoding lengths compared to shorter ones, resulting in less significant performance improvements. For retrieval tasks, ZIGZAGATTENTION achieves an overall high score but still exhibits performance degradation relative to other methods. Nevertheless, these limitations highlight key areas for further analysis and provide a clear direction for future research.

References

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. 2023. Longbench: A bilingual, multitask benchmark for long context understanding. *arXiv preprint arXiv:2308.14508*.

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.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. *arXiv preprint arXiv:1903.00161*.

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*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Greg Kamradt. 2024. Llm test needleinahaystack: Doing simple retrieval from llm models at various context lengths to measure accuracy. https://github.com/gkamradt/LLMTest_NeedleInAHaystack.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Zichang Liu, Jue Wang, Tri Dao, Tianyi Zhou, Binhang Yuan, Zhao Song, Anshumali Shrivastava, Ce Zhang, Yuandong Tian, Christopher Re, et al. 2023. Deja vu: Contextual sparsity for efficient llms at inference time. In *International Conference on Machine Learning*, pages 22137–22176. PMLR.

Leonid Pekelis, Michael Feil, Forrest Moret, Mark Huang, and Tiffany Peng. 2024. Llama 3 gradient: A series of long context models.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, et al. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

Robert Tibshirani. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288.

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

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

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. 2023. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*.