大语言模型长文本处理方法探索

1 项目背景

近年来，大语言模型（LLMs）如 GPT-4、LLaMA 在自然语言处理（NLP）领域取得了显著进展。这些模型在多种任务上表现优异，包括语言生成、翻译、问答系统以及文本摘要等。然而，尽管大语言模型在处理短文本时展现出极高的精度和流畅性，长文本的处理依然面临诸多挑战。

在长文本处理过程中，模型必须能够有效捕捉和理解上下文信息，这对于保证生成内容的连贯性和准确性至关重要。长文本通常包含更复杂的语义结构和多样化的主题内容，因此，如何确保模型在处理长文本时能够保持全局的一致性，并避免信息丢失或重复，是当前研究的一个重要课题。

此外，长文本处理的计算资源需求也是一个不容忽视的挑战。随着输入文本长度的增加，模型的计算复杂度和内存需求呈指数级增长。这使得大规模长文本处理在实际应用中面临技术瓶颈。因此，在不显著增加计算资源的前提下，提升模型处理长文本的能力，成为了当前研究的一个热点。

本项目旨在探索大语言模型在长文本处理方法，提升大模型在长文本生成和理解任务中的表现。同时，本项目评估不同方法在不同应用场景中的适应性。当前，已有诸多研究专注于大语言模型的长文本处理，其中较为前沿的工作包括《Efficient Streaming Language Models with Attention Sinks》[5] 和《LLM Maybe LongLM: Self-Extend LLM Context Window Without Tuning》[3]。

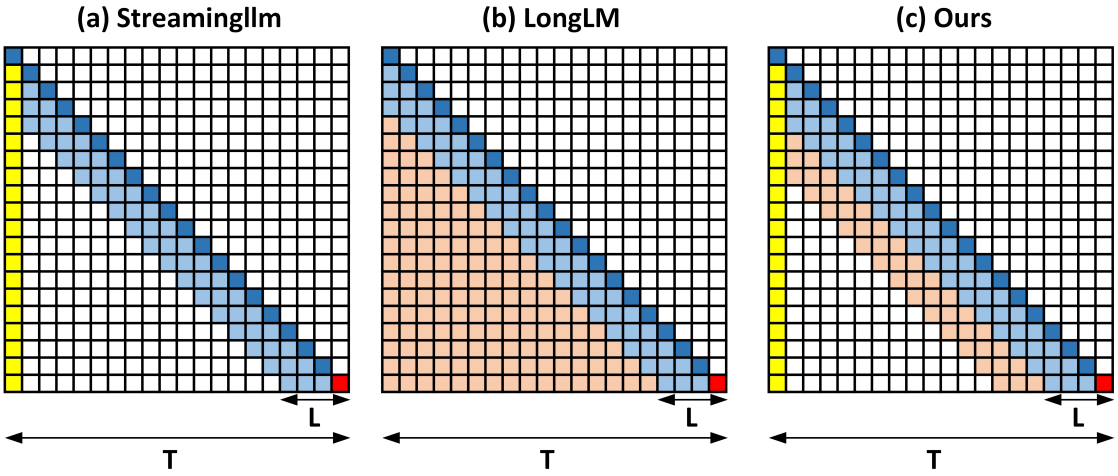

Figure 1: 三种长文本处理方法示意

在《Efficient Streaming Language Models with Attention Sinks》中，研究重点在于大语言

模型的流式输入输出处理。传统的输入方式是一次性将所有数据传递给模型，而流式输入则是逐步将数据传递给模型，这在实时处理和长文本处理中尤为常见。如图1 (a) 所示，StreamingLLM 观察到 Attention Sink 现象，即初始几个 Token 在注意力计算中占有较大比重，对后续 Token 的计算有着关键影响。因此，StreamingLLM 保留了初始 Token，并将窗口长度限定为原始窗口长度。随着文本长度的增加，StreamingLLM 只保留 Attention Sink 与当前窗口长度的信息，而中间部分则被舍弃。这样可以有效实现流式处理，同时避免 KV Cache 过大。然而，这种方法的窗口长度有限，无法胜任一些长文本问答任务。

在《LLM Maybe LongLM: Self-Extend LLM Context Window Without Tuning》中，采用了分组注意力（Group Attention）和邻域注意力（Neighbor Attention）。分组注意力捕捉远距离 Token 之间的依赖关系，而邻域注意力则捕捉在指定范围内相邻 Token 之间的依赖关系。这两级注意力机制在推理过程中基于原始模型的自注意力机制计算。如图1 (b) 所示，LongLM 通过结合分组注意力与邻域注意力，实现了有效的窗口长度扩展。

2 项目设计

本项目中，我们尝试将上述两种方法结合，旨在实现流式处理的同时扩展文本长度。如图1 (c) 所示，我们对 LongLM 中的两级注意力长度进行了缩减，同时保留了最初始的 Token 作为 Attention Sink。

3 项目进展和成果

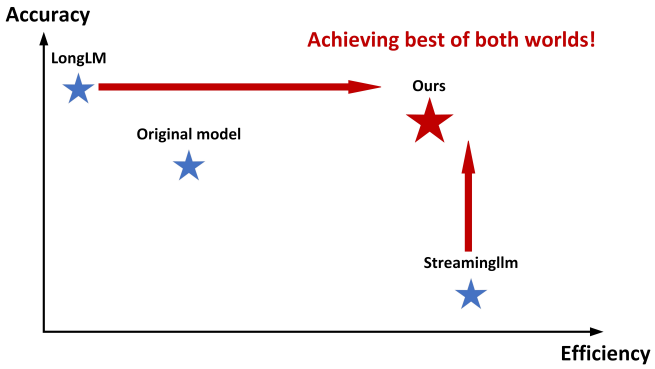

Figure 2: 三种长文本处理方法比较

我们融合了 StreamingLLM 与 LongLM 的实现方式，并采用 LongBench [6] 进行测试，测试模型为 Vicuna-7B-4K。结果如表1所示，从中可以看出，Vicuna-7B-4K 在某些长文本任务上的表现不佳。LongLM 有效扩展了文本长度，在与长文本处理密切相关的任务中，相较于原始模型有所提升；然而，StreamingLLM 在长文本处理方面的表现较差，尤其在长文本问答和信息检索任务中。通过将两种方法相结合，我们的方法实现了文本长度的拓展，并在多种任务中相比 StreamingLLM 取得了提升。

此外，我们还在 MT-Bench [1]（多轮问答）上测试了三种方法。由于 MT-Bench 的文本长度较长，LongLM 很快出现了显存不足的问题；相比之下，StreamingLLM 与我们的方法均能够完成对话，并输出有效结果。

在实际使用大语言模型进行推理的过程中，准确性（Accuracy）和效率（Efficiency）同样重要。在有限的 GPU 显存资源下完成模型推理至关重要。我们在单卡 A100-80G 上展开了测试。

Table 1: 不同方法 Accuracy 比较

Metrics	Original model	LongLM	Streamingllm	Ours
vcsum	4.58	**15.48**	3.93	8.31
dureader	11.95	**22.79**	7.62	12.51
repobench-p	**48.13**	39.52	12.75	15.17
passage_retrieval_en	5.50	**8.50**	2.88	6.50
multifieldqa_en	32.72	**40.90**	16.76	20.61
gov_report	17.56	**29.02**	13.66	21.91
multifieldqa_zh	17.33	**37.22**	3.86	5.22
musique	4.59	**5.08**	2.50	3.38
multi_news	25.22	**26.56**	19.34	20.38
qasper	19.12	**30.00**	4.82	8.26
narrativeqa	13.67	**17.80**	5.75	6.86
lcc	**62.40**	58.25	18.36	18.38
trec	66.00	**71.00**	5.58	3.92
lsht	21.25	**24.00**	0.00	0.00
triviaqa	**84.15**	83.62	10.45	10.73
2wikimqa	17.48	**18.33**	4.54	4.69
samsum	39.91	**42.29**	23.75	22.87
passage_retrieval_zh	0.00	4.50	0.00	**6.25**
qmsum	17.41	**22.28**	15.16	19.02
hotpotqa	19.93	**22.75**	5.84	6.66
passage_count	0.62	2.50	1.75	**3.10**

Table 2: 不同方法 Efficiency 比较

	Original model	LongLM	Streamingllm	Ours
显存占用 (GB)	OOM	OOM	38	66
平均推理时间 (s)	-	-	2.56	14.03

结果如表2所示，在处理长文本时，原始模型和 LongLM 由于 KV Cache 过大，在处理较长的文本时会出现显存不足的问题，无法继续推理，需要借助 VLLM [4]、FlashAttention [2] 等优化方法进行辅助。相比之下，StreamingLLM 采用 Greedy Generate 策略，同时仅保留有限的 KV Cache，在显存占用与推理时间方面表现出色。我们的方法同样有效减少了 KV Cache，实现了单卡推理。

整体测试结果如图2 所示，可以看出我们的方法结合了两种现有方法的优势，在准确性和效率之间达到了平衡。

4 研究创新点

在本研究中，我们研究了已有大语言模型长文本处理方法的优势与不足，尝试将两种各有优劣的方法进行结合，达到了两全其美的效果。

References

[1] 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. Longbench: A bilingual, multitask benchmark for long context understanding. *CoRR*, abs/2308.14508, 2023.

[2] Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

[3] Hongye Jin, Xiaotian Han, Jingfeng Yang, Zhimeng Jiang, Zirui Liu, Chia-Yuan Chang, Huiyuan Chen, and Xia Hu. LLM maybe longlm: Self-extend LLM context window without tuning. *CoRR*, abs/2401.01325, 2024.

[4] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In Jason Flinn, Margo I. Seltzer, Peter Druschel, Antoine Kaufmann, and Jonathan Mace, editors, *Proceedings of the 29th Symposium on Operating Systems Principles, SOSP 2023, Koblenz, Germany, October 23-26, 2023*, pages 611–626. ACM, 2023.

[5] 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, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

[6] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.