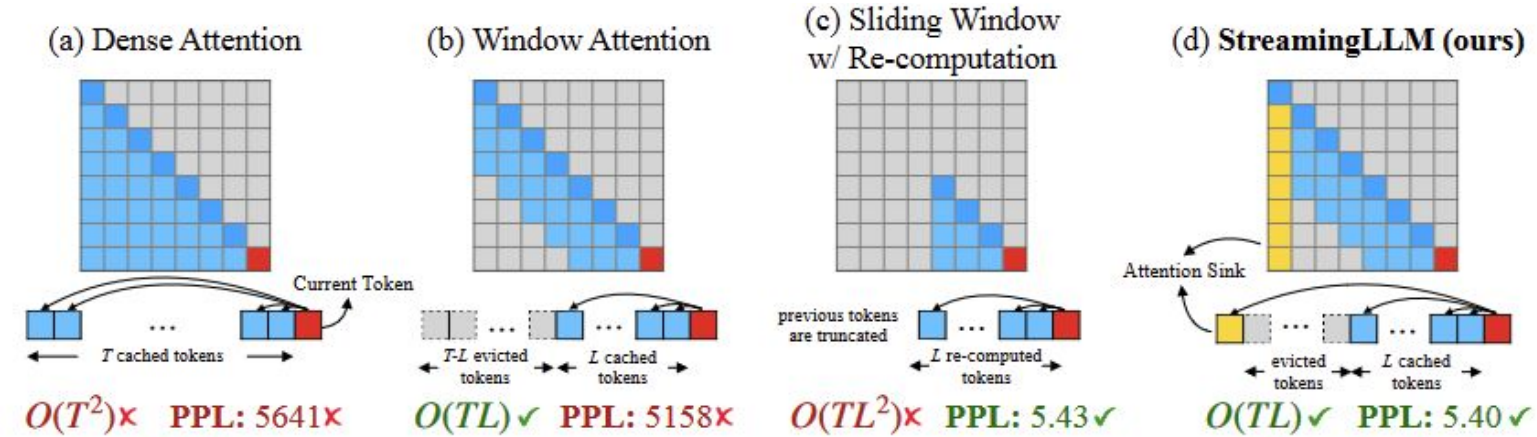
Improving StreamingLLM's Long-Horizon Performance with RAG

Yash Agarwal, David Chaudhari, Nathan Chen, Kenneth Choi, Sameer Pai

Table of Contents

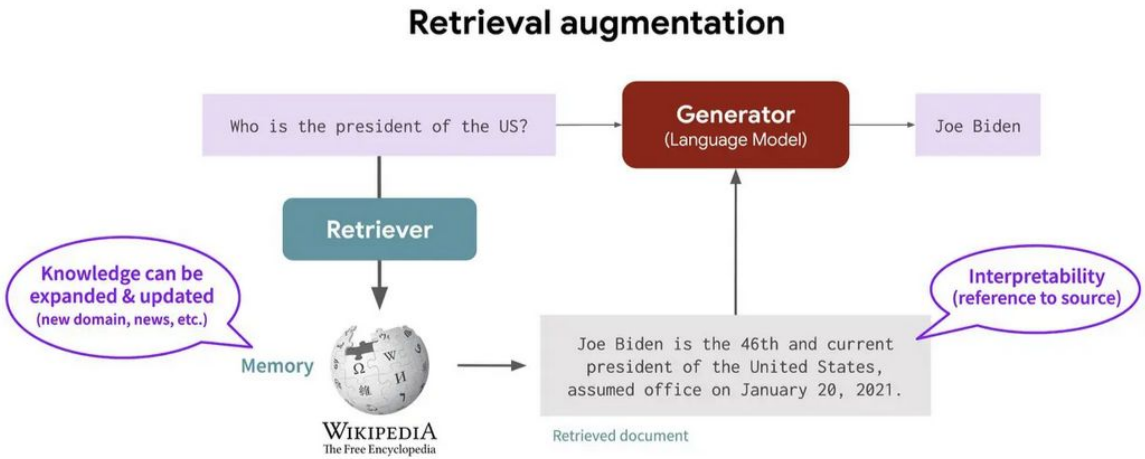
Related Work: StreamingLLM

StreamingLLM improves on the *sliding window KV cache* by also preserving the first few tokens (**attention sinks**). However, the model still can't attend to evicted tokens, leading to memory loss and weakness on tasks such as long document QA and summarization.



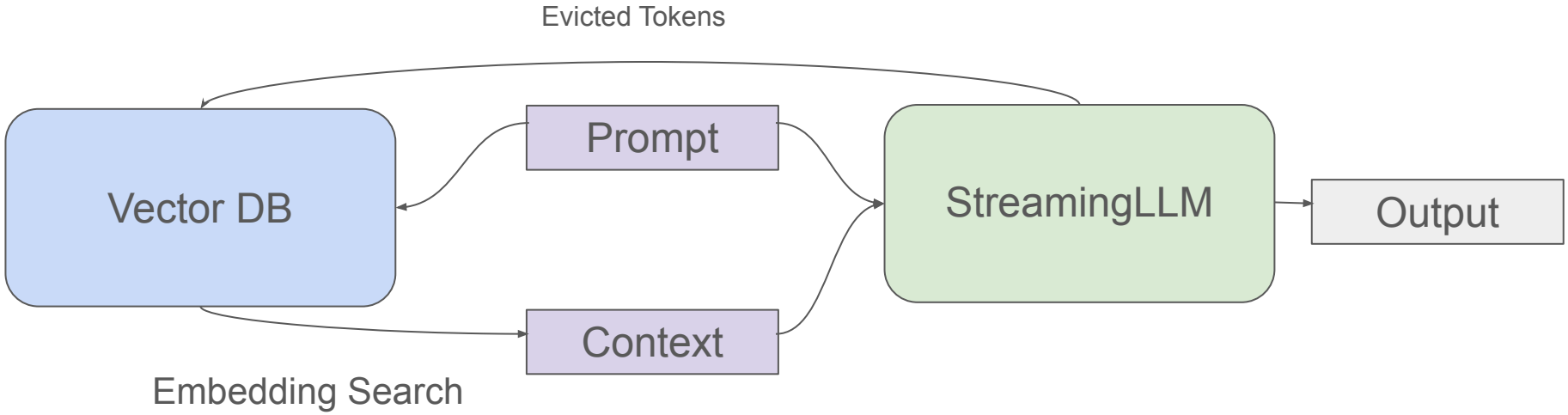
Related Work: RAG

Retrieval-Augmented Generation (RAG) is a technique for improving relevancy of LLM outputs. It works by using a *retrieval system* (e.g. KNN) to identify relevant context from an external knowledge database, and then injecting said context into the LLM prompt.



Method: Fusion of StreamingLLM and RAG

We attempt to improve StreamingLLM's long-context ability by adding evicted tokens to a vector-store database, and then using RAG to augment subsequent user prompts with relevant context from this database.



Example Result

*Using 256 token window for exaggerated effect
*Base model is Llama-3.2-8B-Instruct. Greedy decoding.

USER: I have an appointment at **6pm on October 9th**. Meanwhile, please write an engaging blog post about things to do in New York City.

ASSISTANT: [...]

USER: When is my appointment on October 9th?

Full Attention	StreamingLLM	StreamingLLM + RAG
ASSISTANT: 6pm on October 9th! […]	ASSISTANT: I have a doctor's appointment on October 9th?	SYSTEM: The following text may be relevant [...] ASSISTANT: Your appointment on October 9th is at 6pm. [...]

Evaluation: Short-Term Memory

We evaluate Dense Attention, StreamingLLM, and StreamingLLM w/ RAG on **MT-Bench**. MT-Bench contains daily conversations and short document QA.

We set up our own evaluation suite to evaluate objectively with GPT-4o-mini as a judge instead of using FastChat.

Daily Conversation

USER: Pretend yourself to be Elon Musk in all the following conversations. Speak like Elon Musk as much as possible. Why do we need to go Mars?

ASSISTANT: 1. Why do we need go to Mars?

Hey there, I’m Elon Musk. [...]

Results: MT-Bench

*Base model is Vicuna-7Bv1.3. Greedy decoding.

Evaluation Methods: Long-Term Memory

Our main benchmark will be model performance on long-document Q&A. This is a task that both traditional LLMs and StreamingLLM would struggle on.

Specifically, we will use the **MultiFieldQA** dataset, consisting of long-form scientific articles paired with questions about the article content.

The model's response will be measured with the F1 metric.

Vitamin K - Wikipedia [...]	What are the three synthetic types of vitamin K?	The three synthetic forms of vitamin K are vitamins K3(menadione), K4, and K5
Document	Question	Answer

Results: Document Question Answering Performance

We compare the performance of StreamingLLM + RAG against the performance with just StreamingLLM. The base model here is Llama-3.2-8B-Instruct.

Model	Recall	Precision	F1
StreamingLLM	0.136	0.159	0.136
StreamingLLM+RAG	**0.298**	**0.372**	**0.298**

Evaluation Methods / Results: Speed

We generated 5000 tokens using conversation prompts from **MT-Bench**. Performance costs come from querying the vector store and adding to it.

StreamingLLM	StreamingLLM + RAG
25.20 token/sec	25.04 tokens/sec

*Base Model = Llama3.2-3B-Instruct. Greedy Decoding
*Run on Google Colab L4 GPU Instance

Without RAG, our estimate is that for typical use cases, StreamingLLM runs 1.006x faster than with RAG. This means that StreamingLLM+RAG retains close to the 22x speedup over Sliding Window w/ Recomputation as stated in the original paper

Conclusion

We have developed a method to maintain infinite context length for StreamingLLM, at the slight cost of speed.

This model shows a significant improvement on tasks requiring retention of a large amount of data, such as long-document QA when compared to prior baselines, demonstrating that we've effectively overcome a common shortcoming of finite-context LLMs.

References

[1] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020. 1

[2] Harrison Chase. LangChain, 2022. 1

[3] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library. 2024. 1

[4] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019. 1

[5] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*, 2023. 1

[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, 2023. 1