KVLink: Accelerating Large Language Models via Efficient KV Cache Reuse

Jingbo Yang* [1], Bairu Hou* [1], Wei Wei [2], Yujia Bao [2], Shiyu Chang [1]

1. UC Santa Barbara 2. Center for Advanced AI, Accenture

* Equal Contribution

Motivation

- When building RAG system, context of different queries will be **overlapped**

`Question1:` "When was the Eiffel Tower built?"

Retrieved Document 1: "The Eiffel Tower is located in Paris, France ."
Retrieved Document 2: "Construction of the Eiffel Tower was completed in 1889."

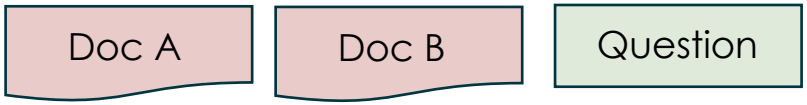
`Question2:` "How tall is the Eiffel Tower? "

Retrieved Document 1: "The Eiffel Tower is located in Paris, France ."
Retrieved Document 2: "The Eiffel Tower stands about 300 meters tall."

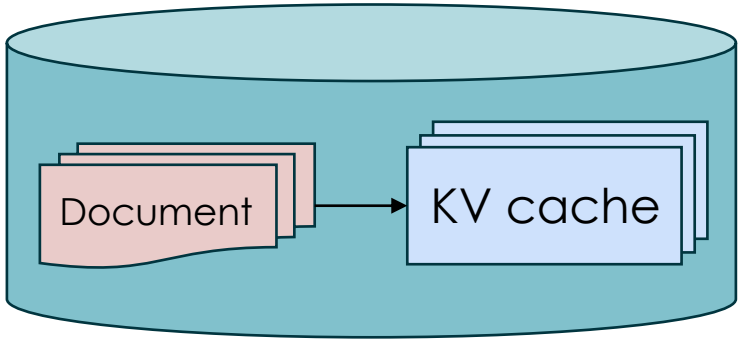
KV Cache Reuse

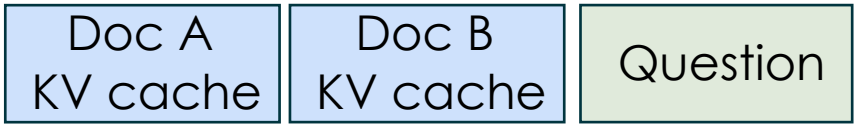
- If we have a pool of KV for each document, we can **directly reuse them.**

Standard decoding



KV reuse

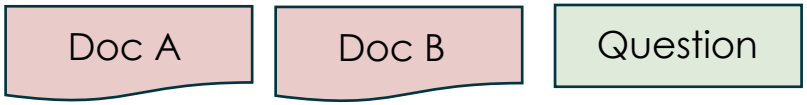




KV Cache Reuse

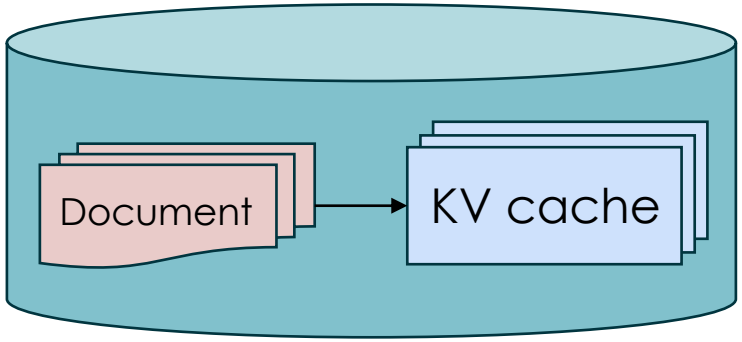
- If we have a pool of KV for each document, we can **directly reuse them.**

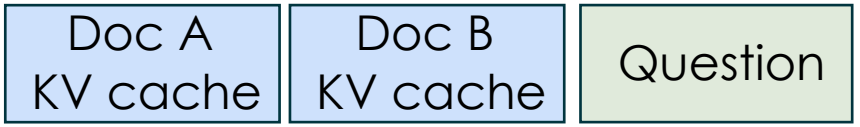
Standard decoding



However, they are not identical.
We want to minimize their discrepancy.

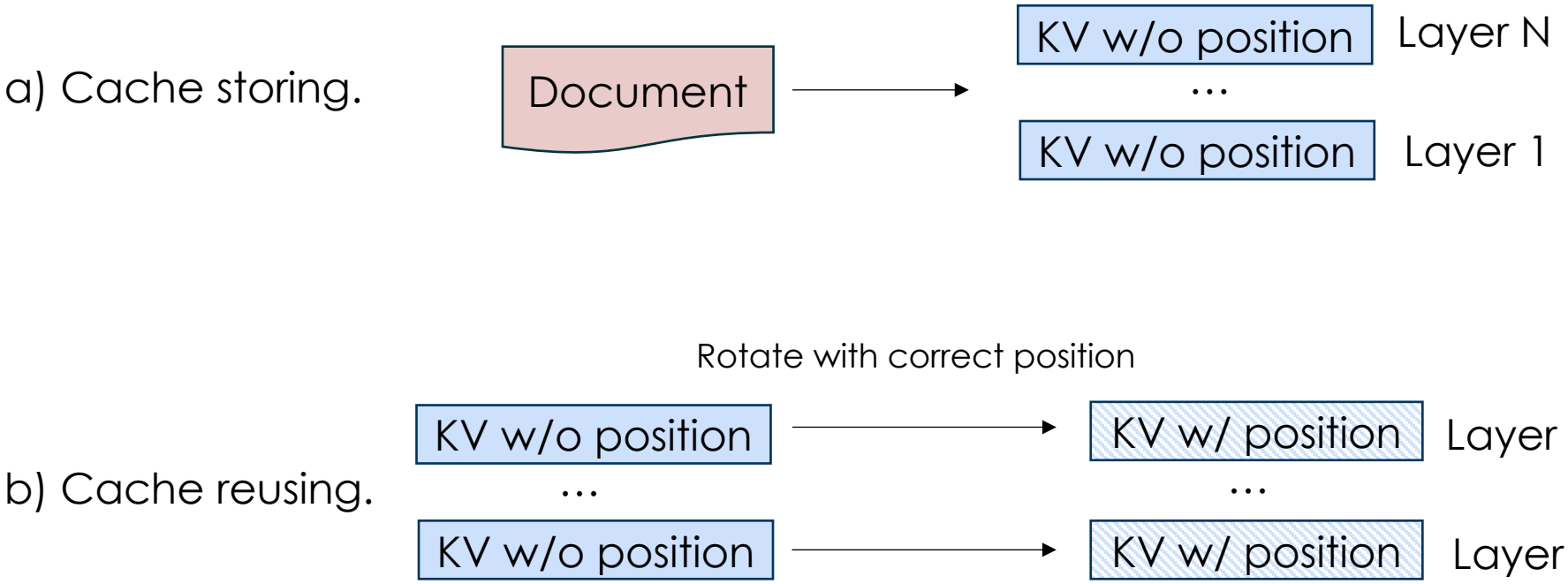
KV reuse





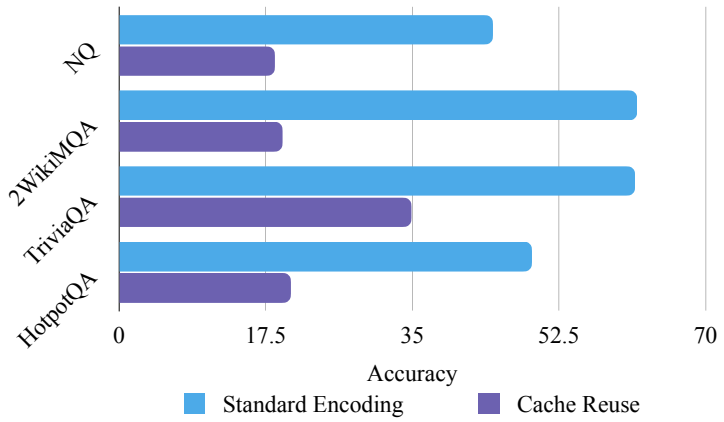
Position Discrepancy

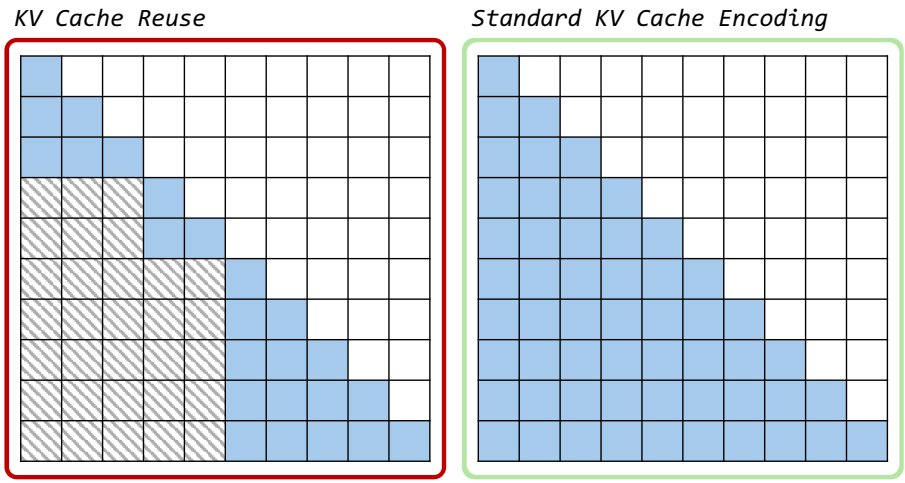
- Retrieved documents can be at any position in the new prompt, so we need to solve position discrepancy first.



Lost Cross Attention

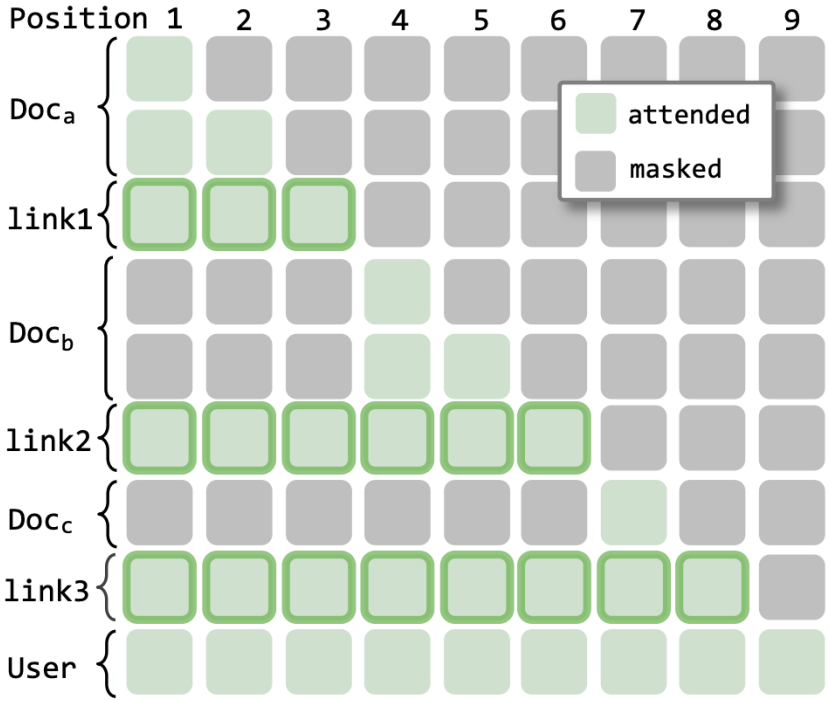
- Although we fix the position discrepancy, we still can observe a huge performance degradation when reusing KV cache.
- This is because, the attention between the documents is lost, compared to the standard decoding.

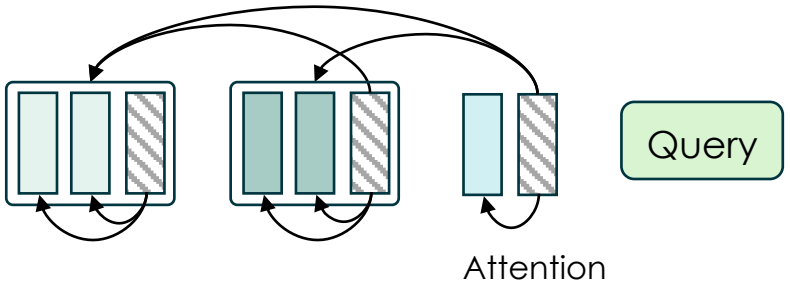




Our solution – inference with KVLink

- We propose adding some special "link" tokens (**with standard attention**) between the documents.





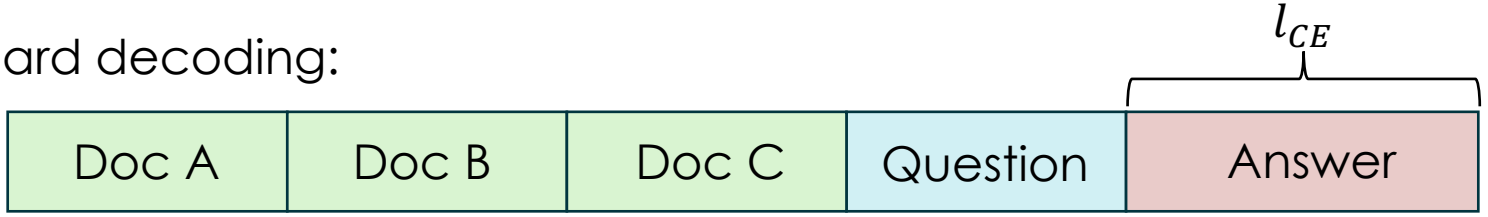
How to train link tokens?

- We find-tuned our model using a mixture of datasets with **two objectives.**

	Separately Encoded KV Cache			Standard Decoding		
Task	**Retrieval-aug. QA**	**Multi-turn conv.**	**Summarization**	**Retrieval-aug. QA**	**SFT**	**Pre-training**
Data Source	TriviaQA, 2WikiMQA	DaringAnteater	XSum	TriviaQA, 2WikiMQA	Tulu3-sft-mixture	Fineweb
Percentage	10%	25%	5%	10%	30%	20%
Total # of Samples	20,000	92,700	17,345	20,000	732,100	10,000,000

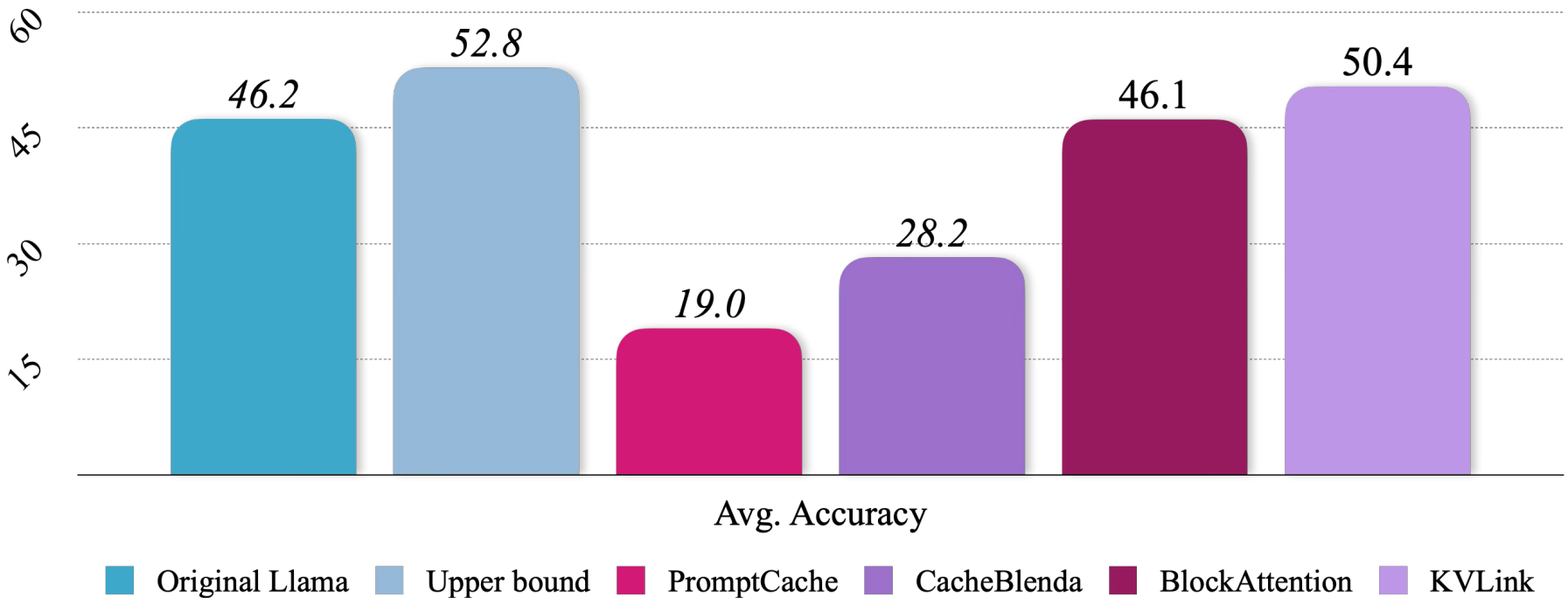
Cache reuse:

Standard decoding:



Results

- KVLink achieves SOTA performance across 5 QA datasets (NQ, 2WikiMQA, TriviaQA, HotpotQA, MuSiQue)



Results

- Pre-filling time with a 5,000-token context using Llama 3.1-8B model