DuoAttention: Efficient Long-Context LLM Inference with Retrieval and Streaming Heads

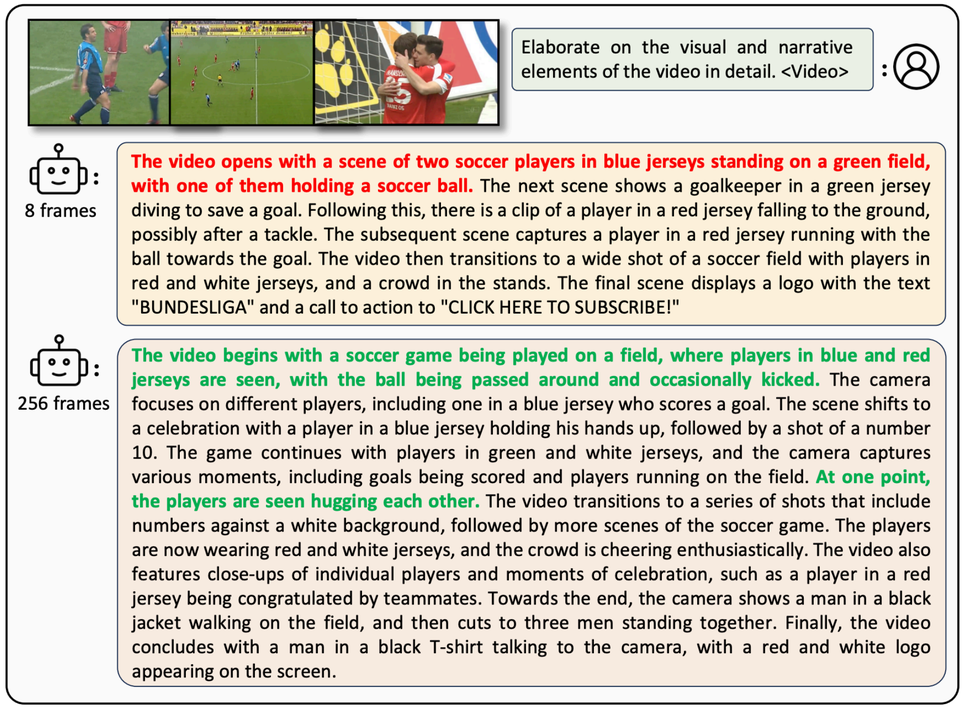
Guangxuan Xiao, Jiaming Tang, Jingwei Zuo, Junxian Guo,
Shang Yang, Haotian Tang, Yao Fu, Song Han

Massachusetts Institute of Technology
Tsinghua University
University of Edinburgh
NVIDIA

Motivation

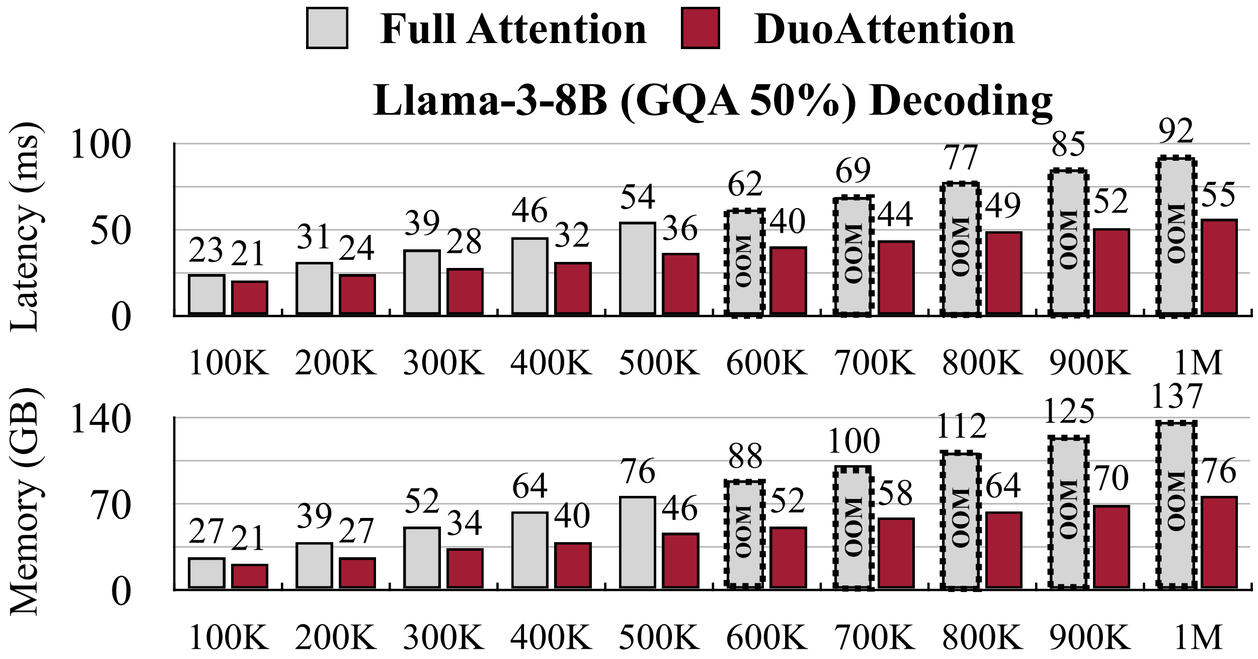
Deploying Long-Context LLMs is Crucial But Challenging

- LLMs need to handle long-context like summarizing long texts and processing images/videos.
- Memory and latency increase dramatically with context length.



a 224×224 image = 256 tokens

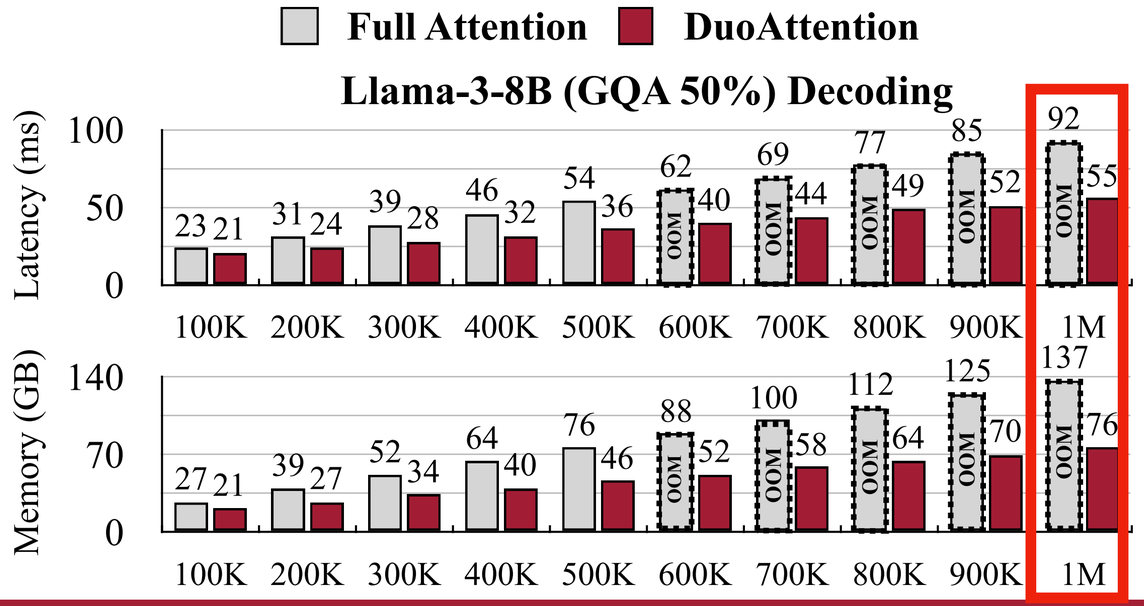
a 1-hour video at 1 FPS = **1 million tokens**

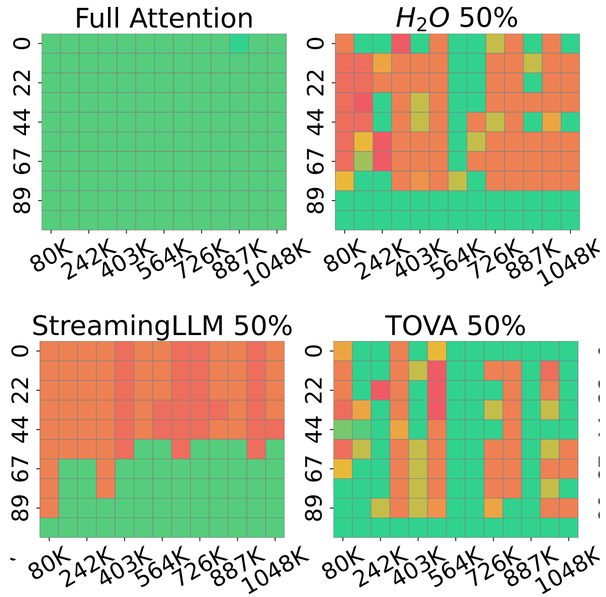


Key Challenges

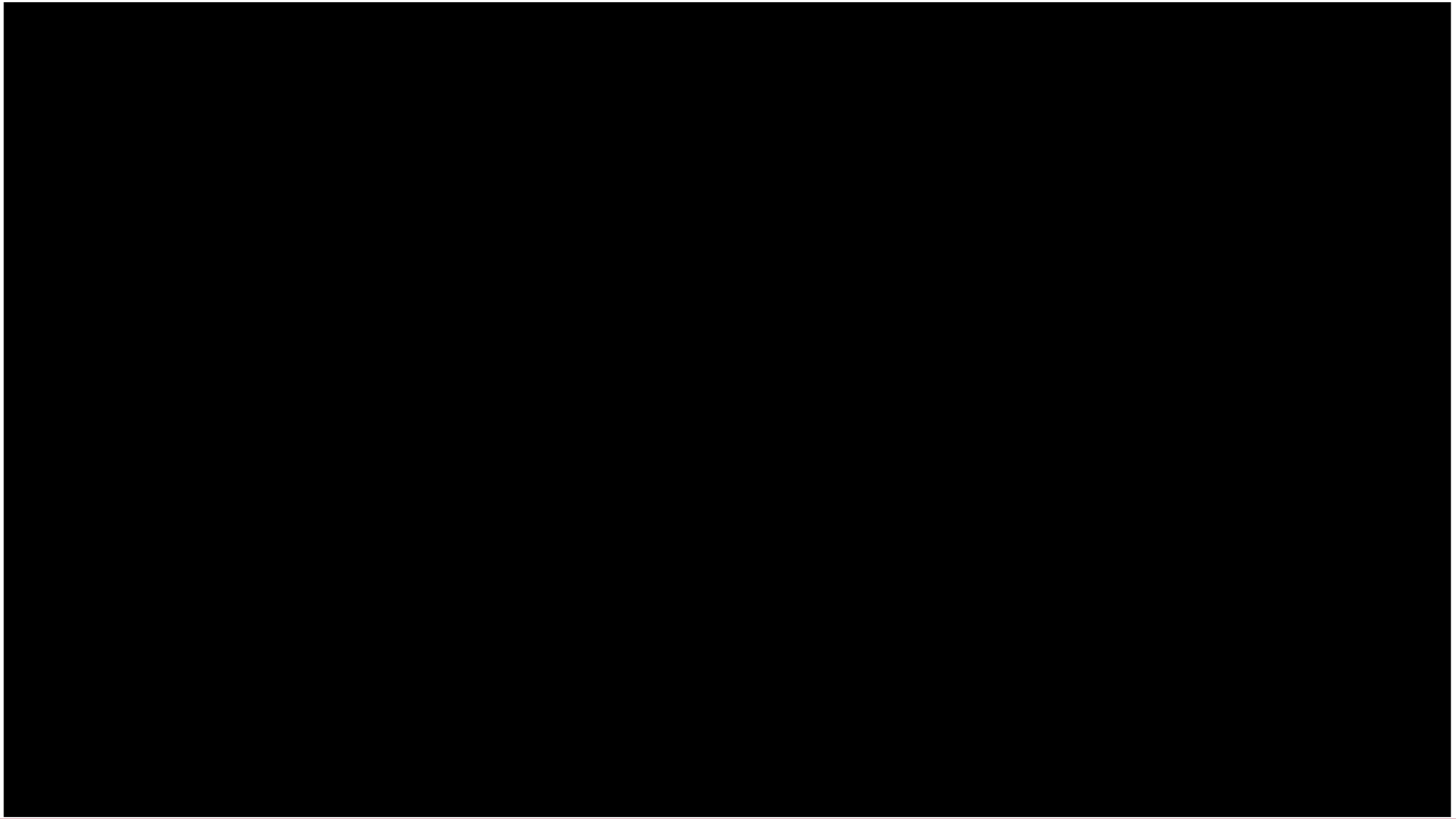
Memory, Latency, and Long-Context Accuracy

- **Memory Bottleneck:** Storing KV states across all tokens consumes massive memory (e.g., 137 GB for 1 million context in Llama-3-8B).
- **Decoding Latency:** Decoding time grows linearly with sequence length (e.g., 92 ms per token for 1 million context in Llama-3-7B).
- Existing KV cache compression methods **damages** LLMs' long-context ability.





Running LLMs with 3.3 Million Contextual Tokens on an A100 GPU

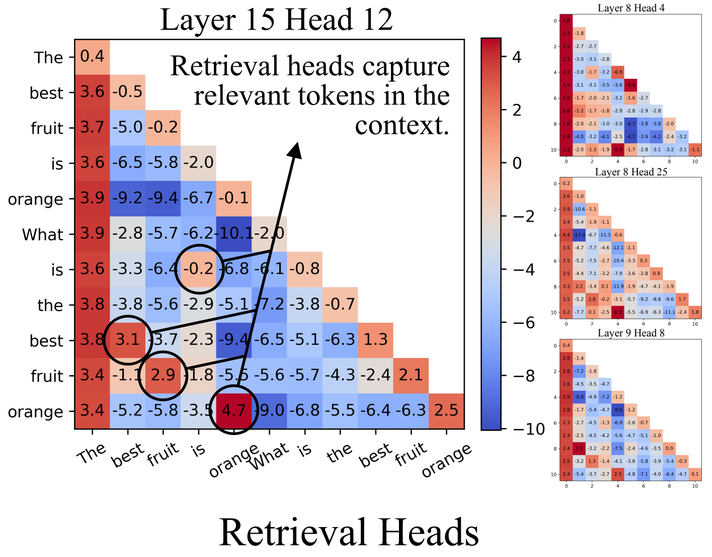


Retrieval vs. Streaming Heads

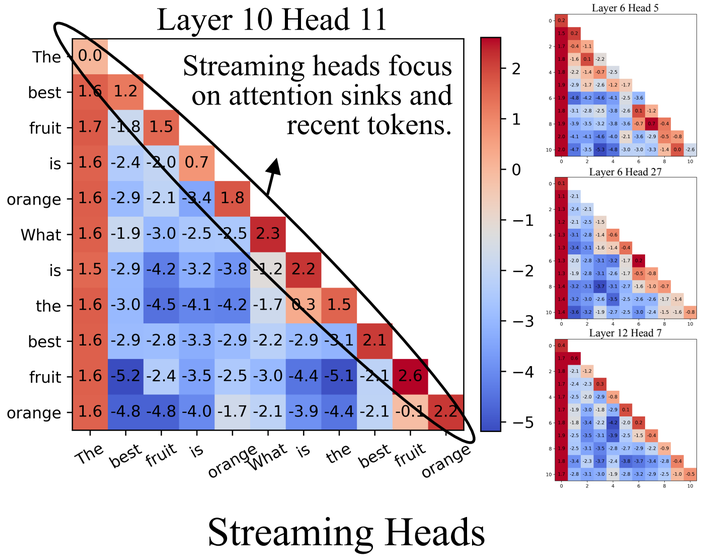
Retrieval Heads vs. Streaming Heads

- **Retrieval Heads:**
  - Capture contextually important tokens from earlier in the sequence.
  - Require full attention across all tokens in the context.
  - Compressing their KV cache would cause a significant loss in performance.

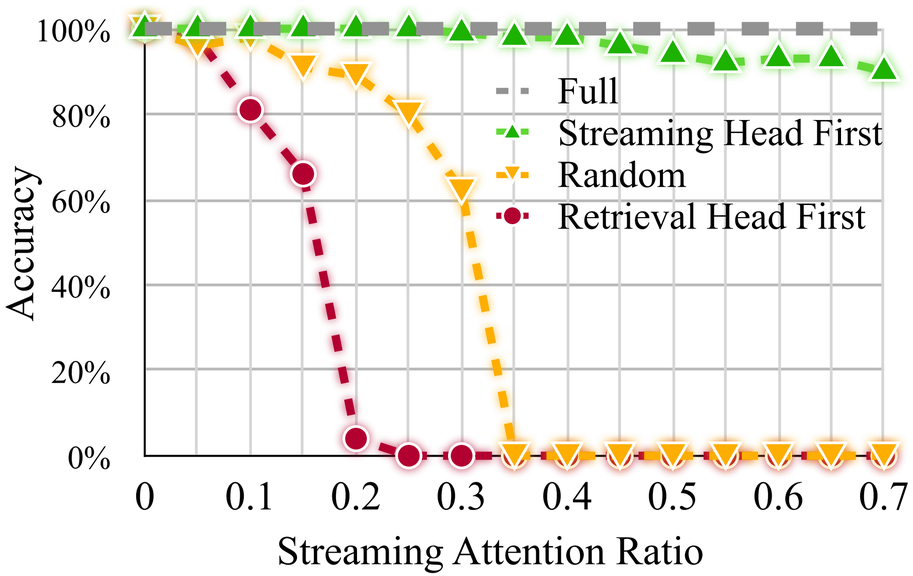
- **Streaming Heads:**
  - Focus on recent tokens and attention sinks.
  - Use a reduced KV cache that only stores recent tokens.
  - Reducing their cache size has minimal impact on performance.



Retrieval Heads



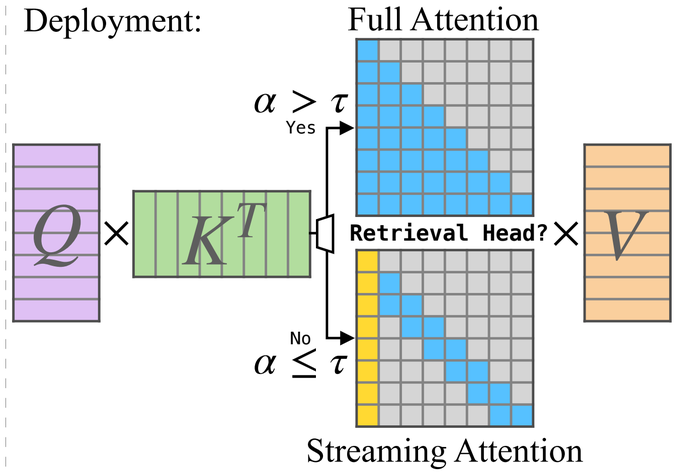
Streaming Heads

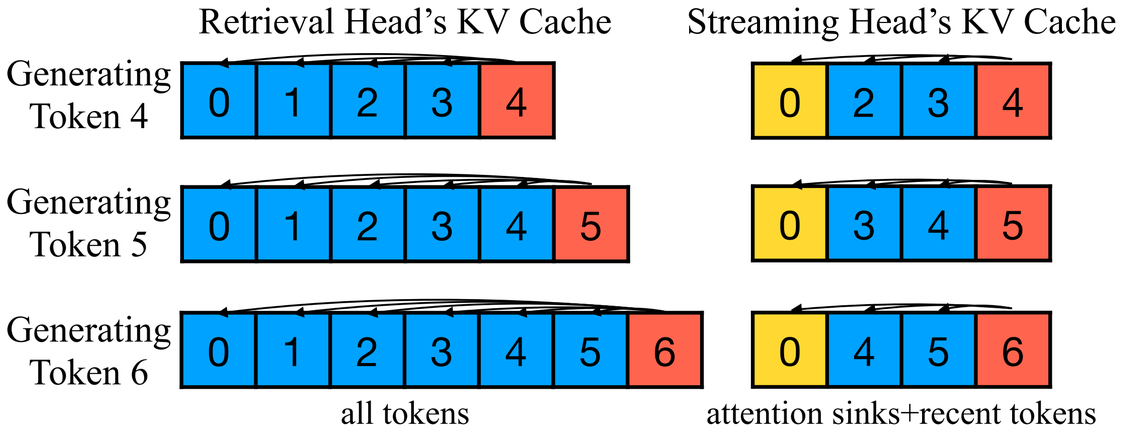


Streaming Attention Ratio

DuoAttention Framework Overview

- **Key Insight: Not all attention heads in a model need full attention across the entire context.**
  - **Retrieval Heads** focus on important tokens from earlier in the sequence and need full KV cache.
  - **Streaming Heads** focus on recent tokens and attention sinks, needing only a reduced KV cache.
- **Solution**: DuoAttention only applies full KV cache to Retrieval Heads, and uses lightweight, constant-length caches for Streaming Heads. Reduces memory and decoding latency while preserving long-context capabilities.



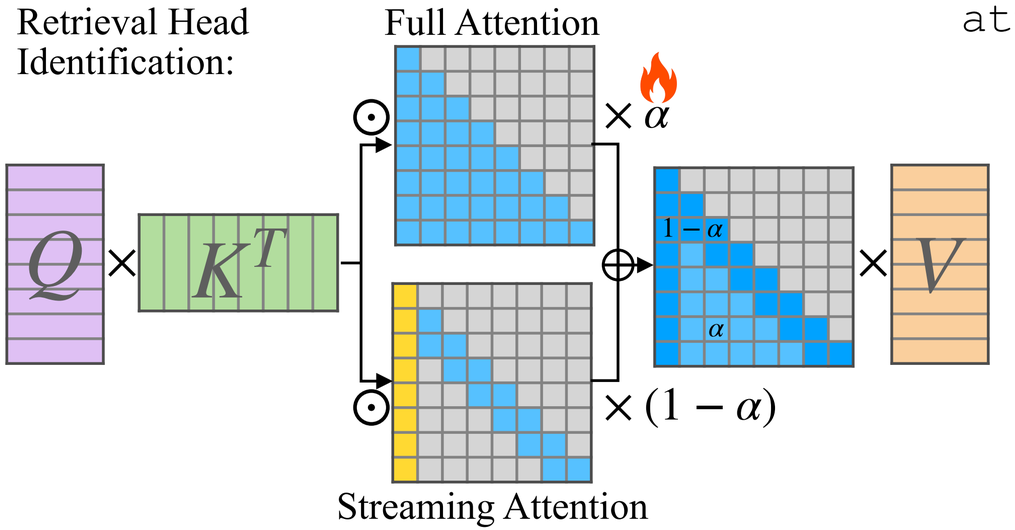


DuoAttention: Efficient Long-Context LLM Inference with Retrieval and Streaming Heads [Xiao et al., 2024]

Optimization-Based Identification of Retrieval Heads

Efficiently and Accurately identify Retrieval Heads

- **Challenge:** Precisely identifying which heads are critical for long-context processing is difficult.
- **Optimization-Based Approach:**
  - Assign a trainable gate value (α) to each head.
  - The gate value blends full attention and streaming attention, allowing the model to learn which heads are necessary.
  - Minimize output deviation from the full attention model to maintain accuracy.



$$\texttt{attn}_{i,j} = \alpha_{i,j} \cdot \texttt{full\_attn} + (1 - \alpha_{i,j}) \cdot \texttt{streaming\_attn}$$

$$\texttt{full\_attn} = \text{softmax}(\boldsymbol{Q}\boldsymbol{K}^T \odot \boldsymbol{M}_{\text{causal}})\boldsymbol{V},$$

$$\texttt{streaming\_attn} = \text{softmax}(\boldsymbol{Q}\boldsymbol{K}^T \odot \boldsymbol{M}_{\text{streaming}})\boldsymbol{V},$$

$$\mathcal{L}_{\text{distill}} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=T-l+1}^{T}(\boldsymbol{H}_{\text{full}}^{(i)}[j] - \boldsymbol{H}_{\text{mixed}}^{(i)}[j])^2$$

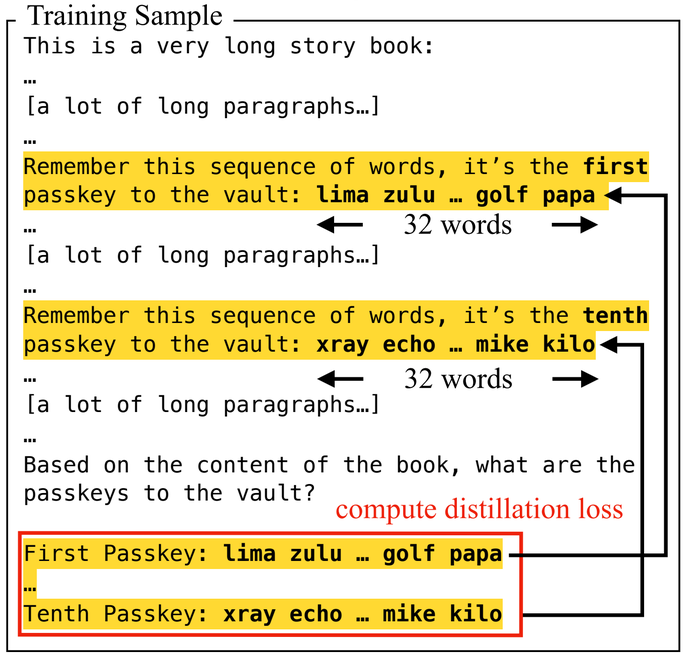
$$\mathcal{L}_{\text{reg}} = \sum_{i=1}^{L}\sum_{j=1}^{H}|\alpha_{i,j}|$$

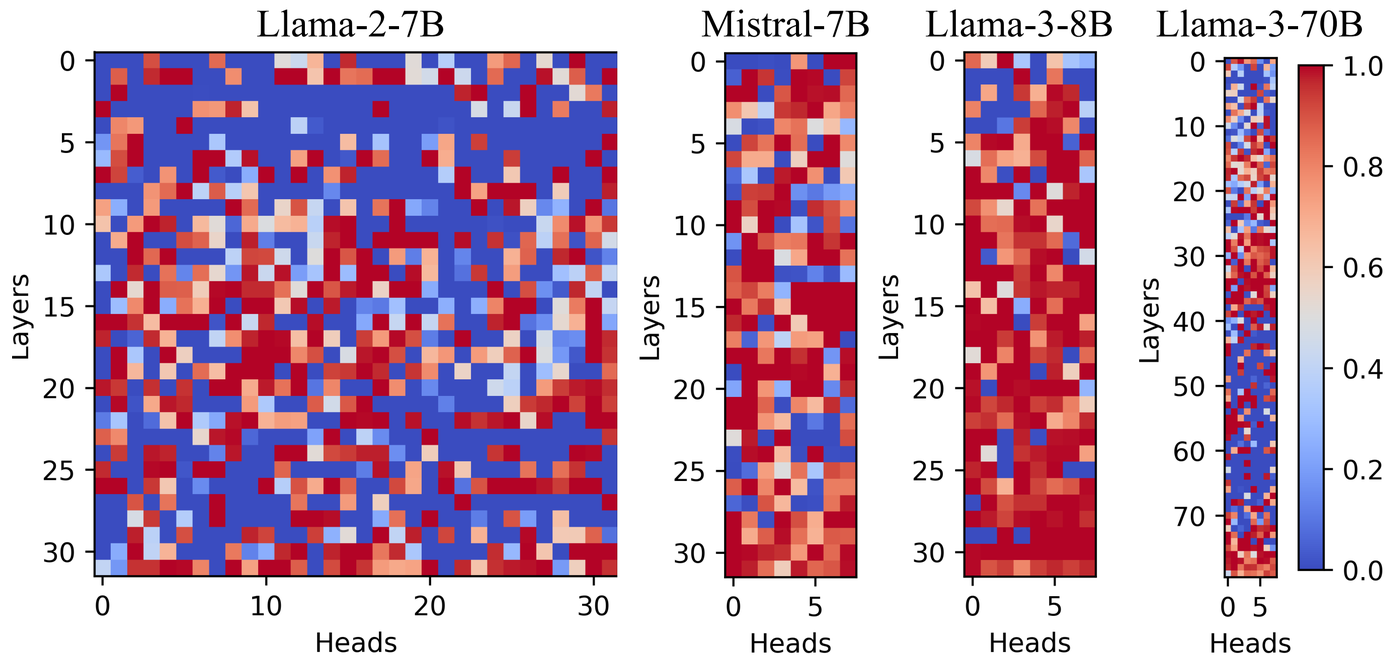
$$\mathcal{L} = \mathcal{L}_{\text{distill}} + \lambda\mathcal{L}_{\text{reg}}$$

Optimization-Based Identification of Retrieval Heads

Using Synthetic Data to Focus on LLMs' Retrieval Ability

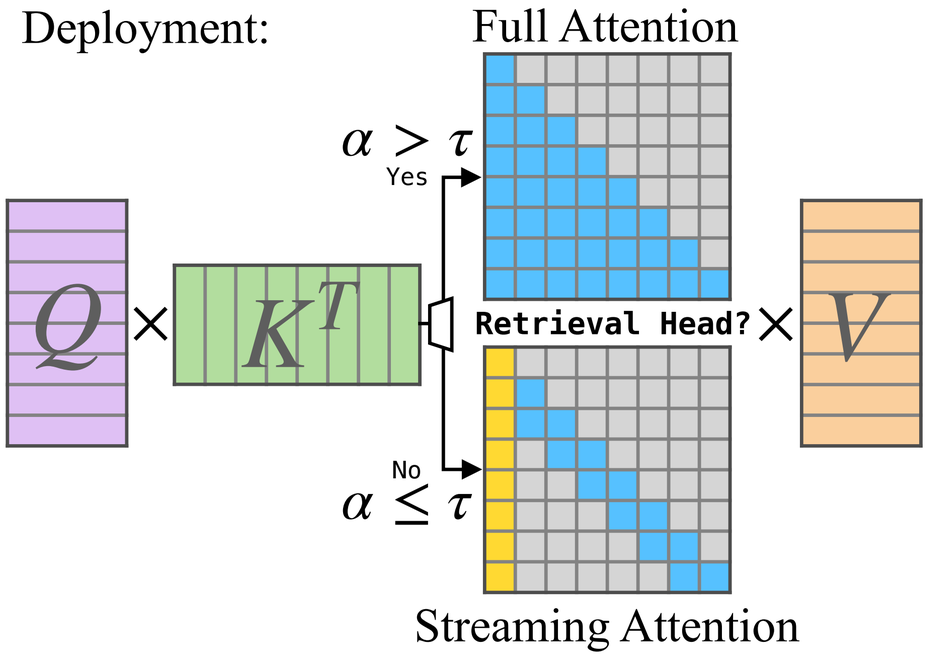
- **Synthetic Dataset:**
  - Embeds ten passkeys in a long context.
  - LLM recalls the passkeys to identify which heads need full attention for long-context retrieval.
- **Efficiency:**
  - Only ~1K gate values need to be trained. (e.g. 32 layers x 32 heads for llama2-7B)
  - The process is completed in a few hours on 8 A100 GPUs.

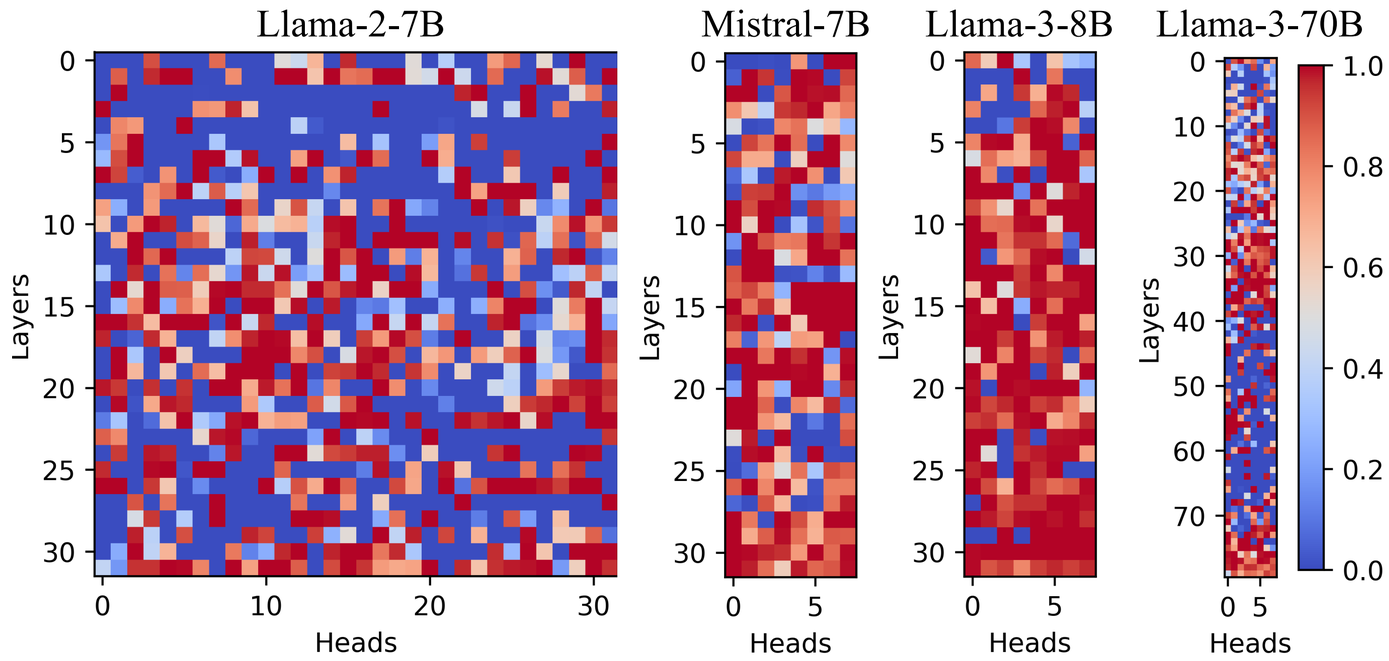




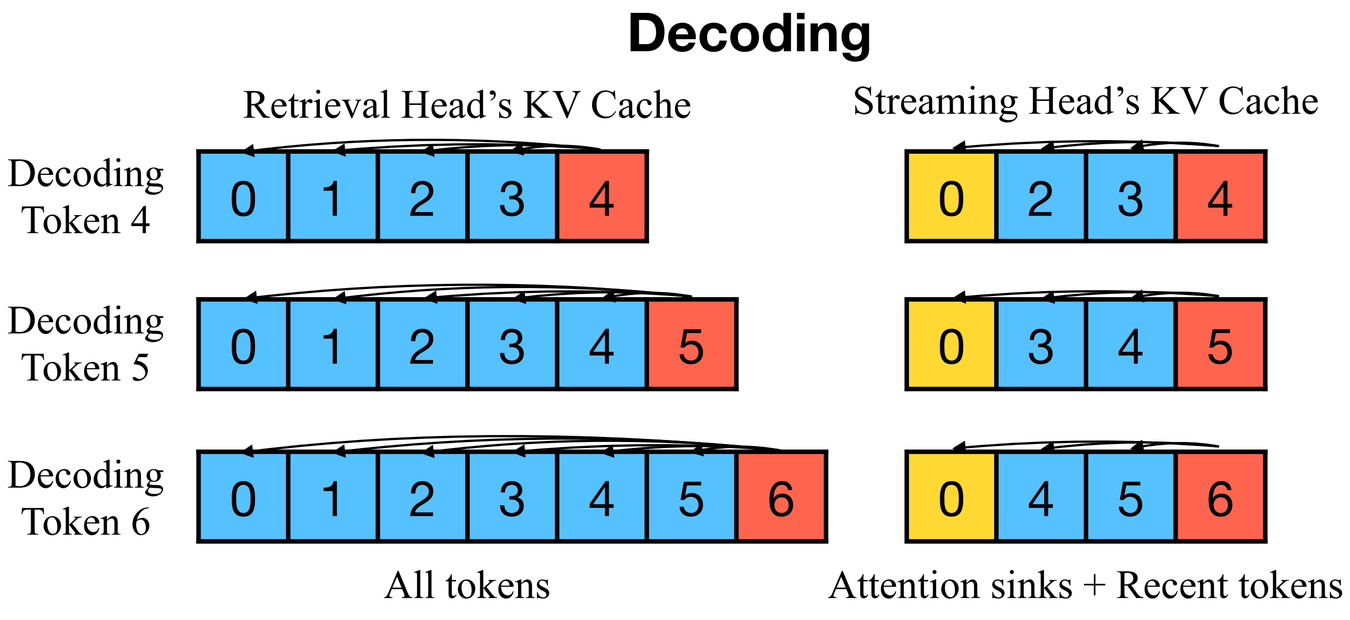
Deploying LLMs with DuoAttention

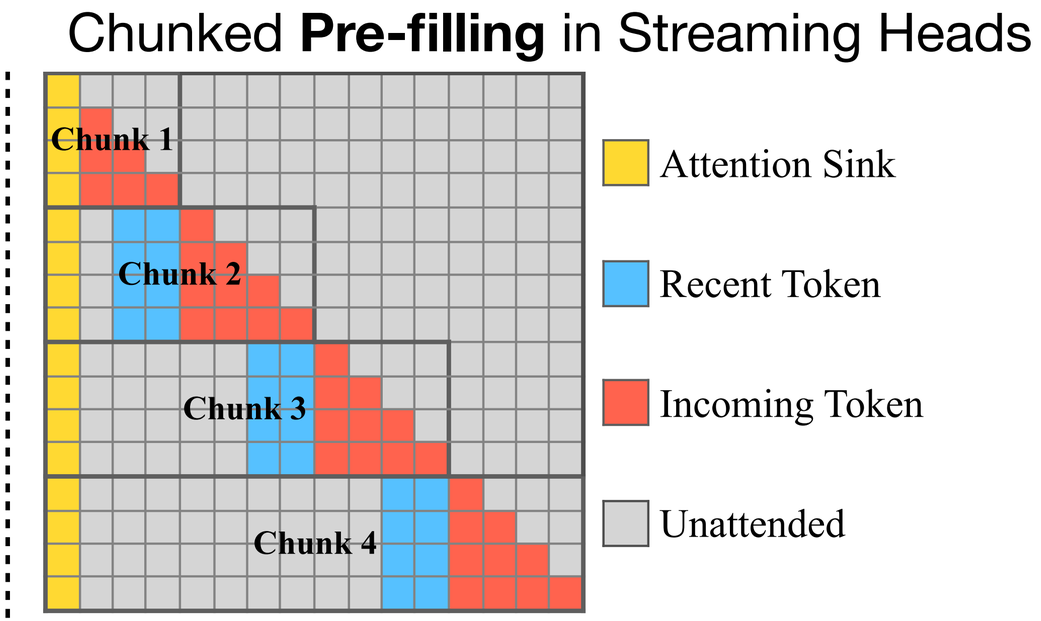
- **Binarizing Attention: During deployment, the trained gate values (α) are binarized to classify each head as either a Retrieval Head or a Streaming Head.**
- **Reordering Attention Heads:**
  - During deployment, attention heads are reordered into distinct clusters for efficient processing.
  - This allows for better slicing and concatenation of KV caches, improving inference speed.





Decoding and Chunked Pre-filling of DuoAttention





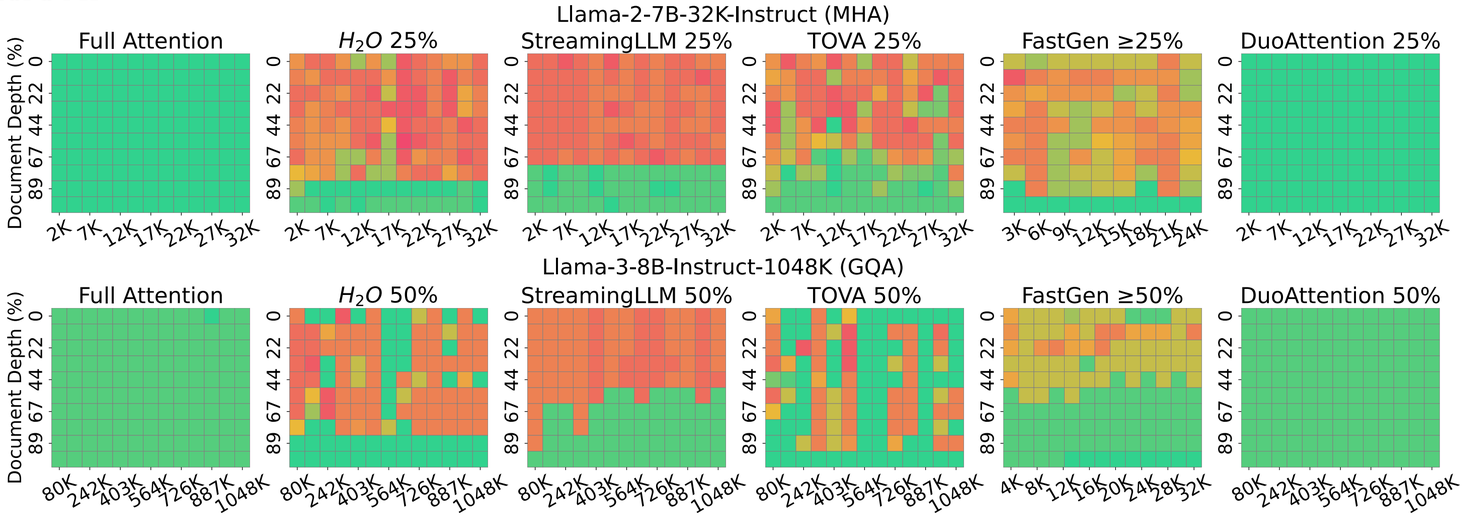
- **Two KV Caches:**
  - **Retrieval Head:** Stores all tokens.
  - **Streaming Head:** Stores attention sinks and recent tokens (constant memory).
- **Computation:**
  - Queries, keys, and values are split into retrieval and streaming heads.
  - Outputs from both heads are concatenated for the final projection.

- **Chunked Pre-filling splits sequence into fixed-length chunks.**
- **Streaming head's KV cache keeps only sink and recent tokens after each pre-filling chunk.**
- **Pre-filling Complexity on Streaming Heads:**
  - Time complexity reduced from $O(L^2)$ to $O(LK)$.
  - Memory complexity reduced from $O(L)$ to $O(K)$
  - L is sequence length and K is chunk size.

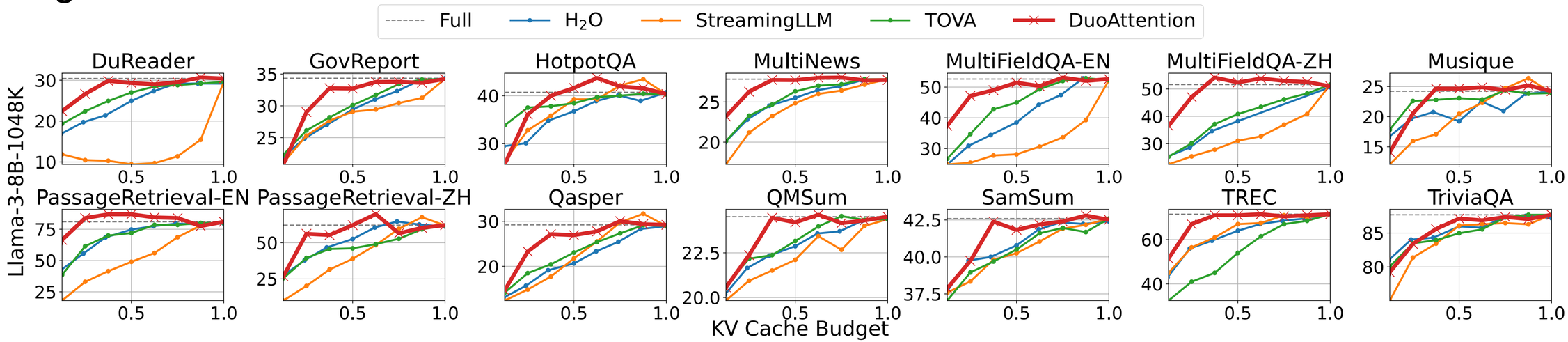
Results on Accuracy Benchmarks

Long-Context Benchmarks: Needle-in-a-Haystack and LongBench

- **Needle-in-a-Haystack:**



- **LongBench:**



Results on Accuracy Benchmarks

Short-Context Benchmarks: MBPP, MMLU, and MT-Bench

- **DuoAttention doesn't harm LLMs other general abilities.**

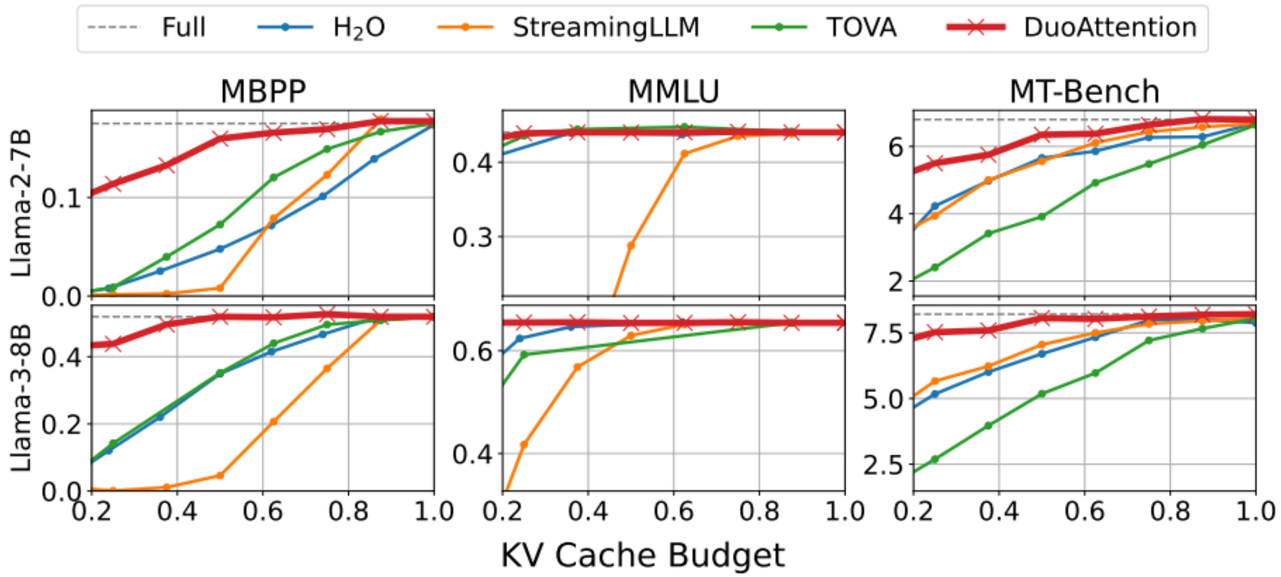


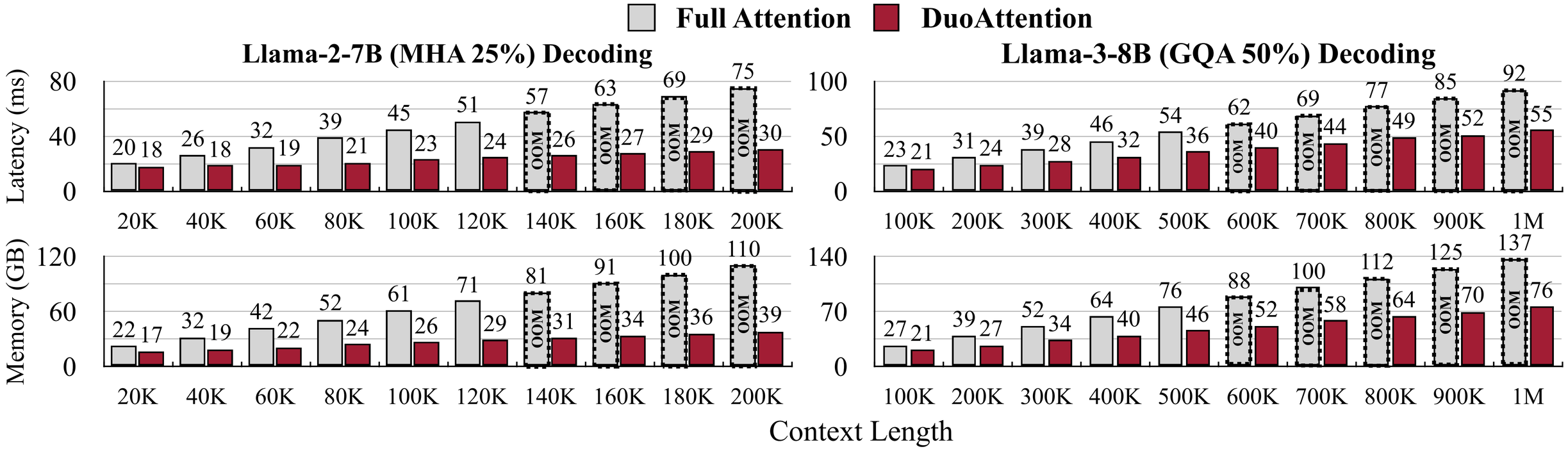
Figure 8: Results on short benchmarks.

Table 1: Llama-3-70B results on short benchmarks.

	Budget	MMLU	MBPP	MT-B
Full	100%	79.38%	47.85%	8.93
H2O	50%	79.26%	32.12%	7.16
TOVA	50%	79.15%	36.09%	7.96
SLLM	50%	77.46%	5.57%	5.41
DuoAttn	50%	**79.35%**	**47.09%**	**9.14**

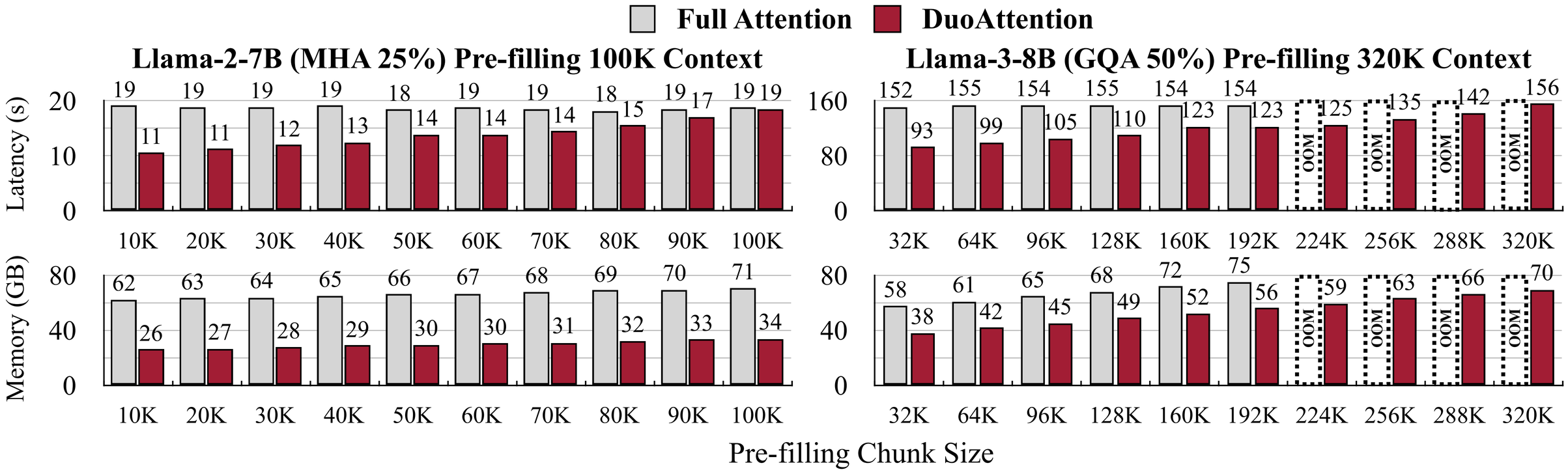
Decoding Memory and Latency Improvements

- **DuoAttention provides up to 2.45× memory reduction for MHA and 1.65× for GQA models, and up to 2.13× decoding latency improvement for MHA and 1.5× for GQA models.**



Pre-filling Memory and Latency Improvements

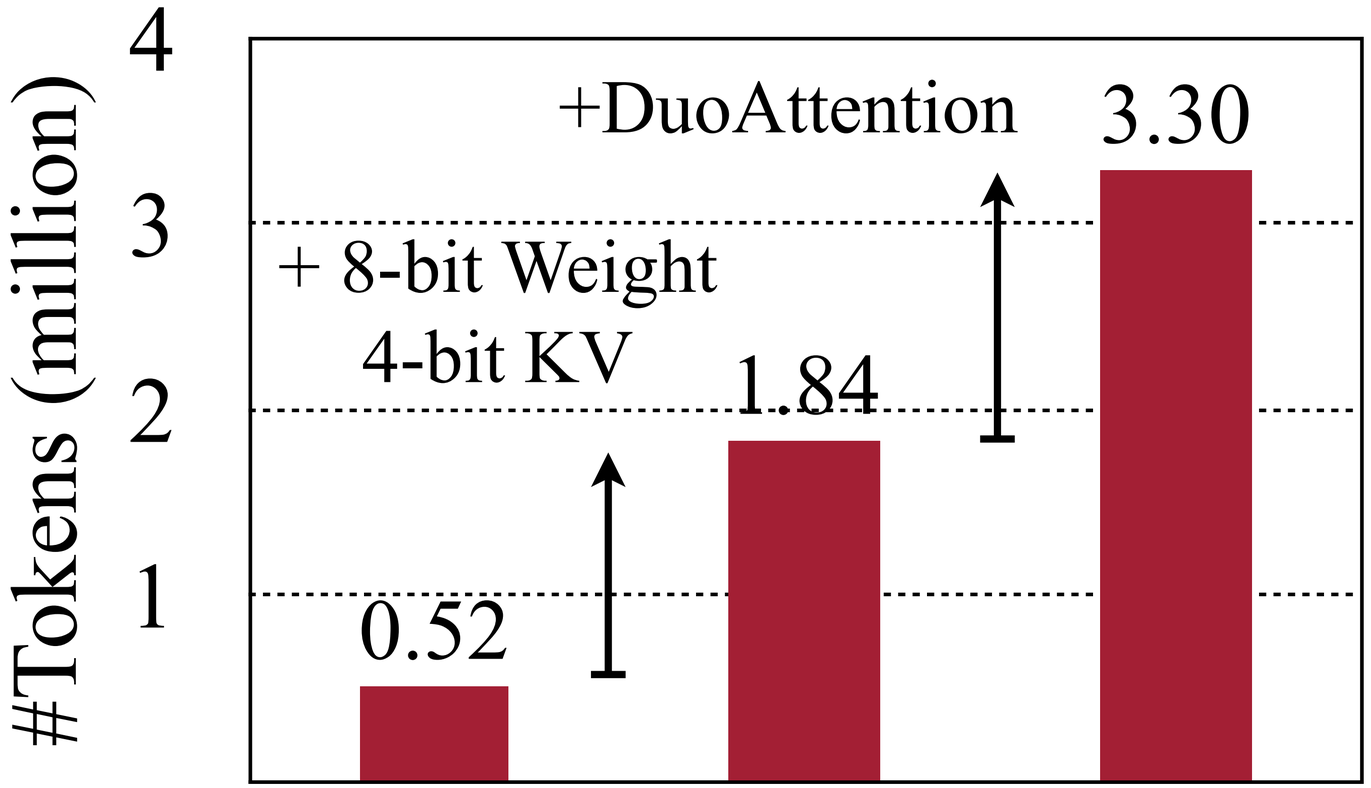
- **DuoAttention achieves up to 1.73x pre-filling latency reduction for MHA and 1.63x for GQA models, with memory reductions up to 2.38x for MHA and 1.53x for GQA models.**



Combination with KV Cache Quantization

Pushing the Limits of Context Length on Memory-constrained Devices

- **Up to 3.3 million tokens can be handled in a single A100 GPU using DuoAttention combined with 4-bit KV cache quantization.**



Conclusion

- **We introduce DuoAttention, an efficient framework that significantly reduces deployment memory and latency in long-context LLMs while maintaining their long-context ability.**
- **Paper:** https://arxiv.org/abs/2410.10819
- **Code:** https://github.com/mit-han-lab/duo-attention
- **Demo:** https://youtu.be/tyTkZOqKt6U