Composing KV Cache Compression Techniques for Long-Context LLM Inference

Huachun Hirairi, Joshua Hong, Hong Lin
{hhirairi, jjhong, honglin}@andrew.cmu.edu
Carnegie Mellon University

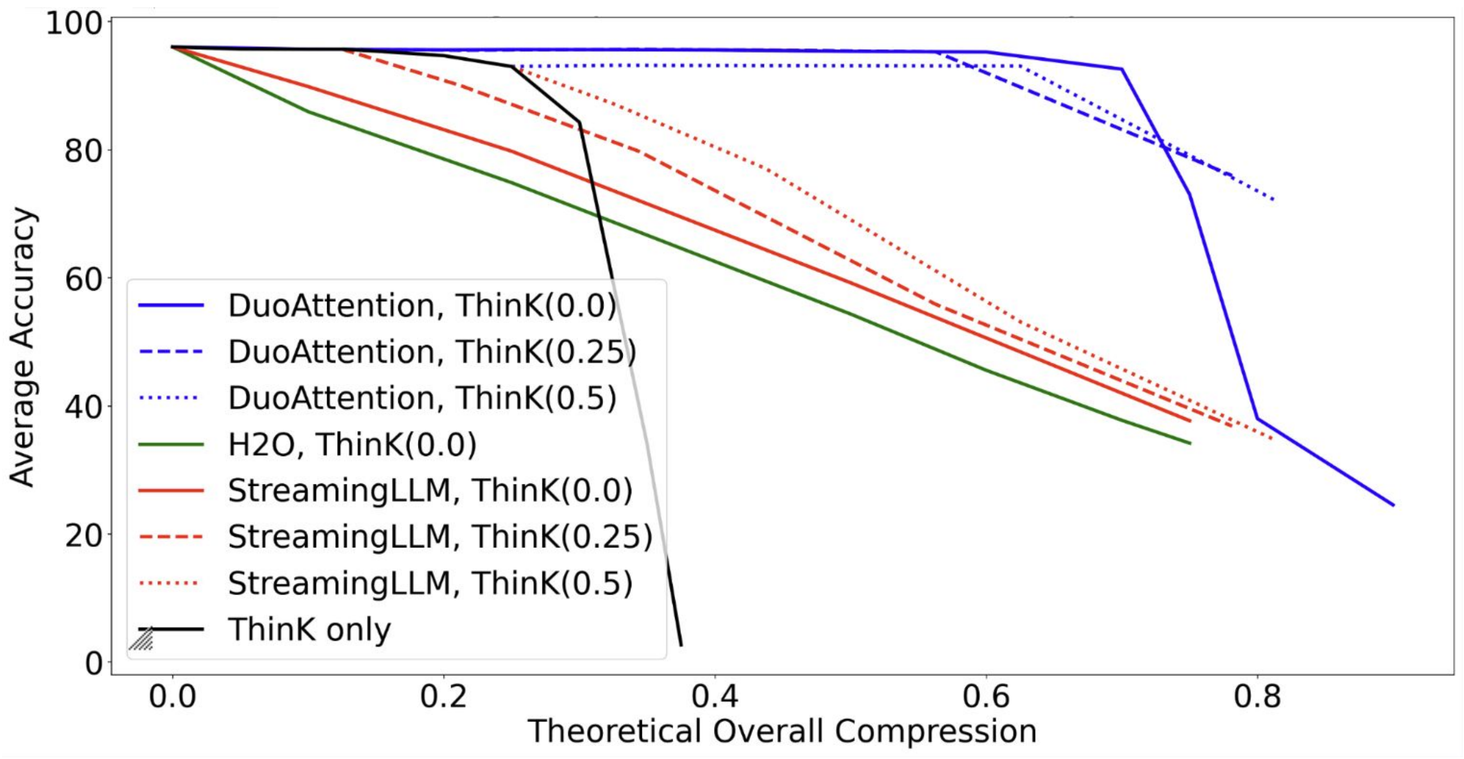


Figure 1: Comparison of accuracy dropping curve when increasing overall compression level of KV cache for 3 presses and their composition with ThinK.

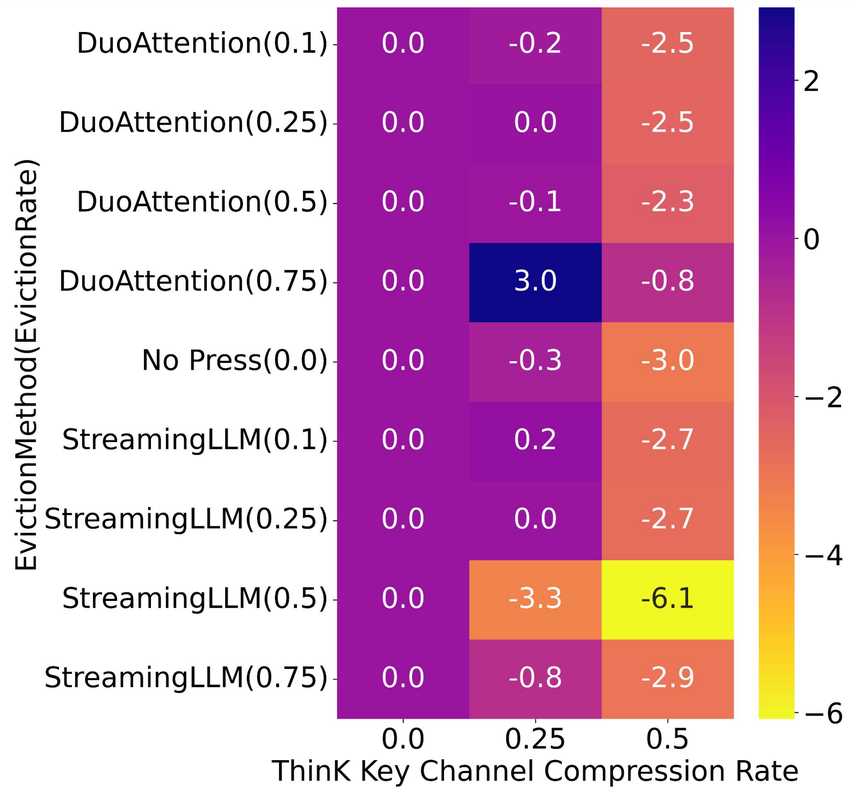


Figure 2: Impact of ThinK composition on accuracy

Introduction

- Various KV cache compression techniques exist that can reduce memory consumption with long sequences.
- We investigate, compare, and compose KV compression techniques across different compression levels.
- We find that further compression with minimal accuracy drop can be achieved by composing orthogonal presses.

Problem Statement

- KV-cache growth limits inference over long sequences, especially under resource-constrained environments.
- Standalone compression methods incur unacceptable accuracy drops when the compression rate is high.

Related Work

Eviction-based Compression

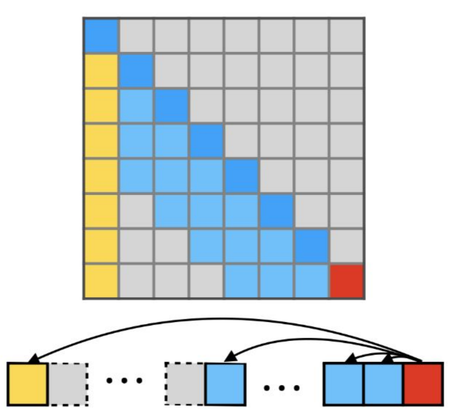
1. **StreamingLLM**: retains initial and recent tokens.
2. **H2O**: retains tokens w/ high averages attention weights.
3. **DuoAttention**: combine retrieval and streaming heads.

Dimensionality Reduction

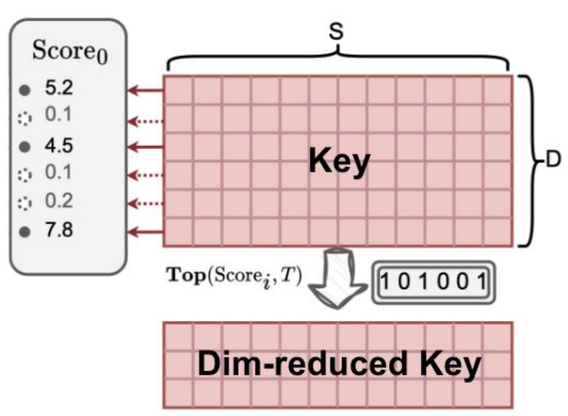
4. **ThinK**: retains key's channels w/ high attention scores.

Other Techniques

5. **Selection-based:** E.g. QUEST, TidalDecode
6. **Merger-based:** E.g ToMe, D2O
7. **KV Cache Quantization**



StreamingLLM



ThinK

Method

- Used NVIDIA/KVPress to compose and evaluate eviction techniques, such as StreamingLLM and DuoAttention, with ThinK, a key-cache dimensionality reduction technique.
- Evaluated long-context capabilities of presses with the RULER 4K benchmark across 13 different long context tasks. All experiments used the pretrained Llama 3.1 8B Instruct model running on PSC Bridges-2.

Evaluation and Analysis

- Fig. 1 shows that composition may achieve higher accuracy vs. standalone
  - Presses behave differently with high compression: sharp accuracy cliff (ThinK, DuoAttention) vs. steady decline (H2O, StreamingLLM)
  - Under high compression, composition with ThinK ⇒ higher accuracy.
- Fig. 2 shows that composing presses at levels before their respective 'accuracy cliffs' (e.g. ThinK(0.25) with DuoAttention(≤0.5)) could allow for high levels of compression with minimal decrease in accuracy.
- Fig. 3 and 4 suggest that the effects of compression are not consistent across task types.
  - Aggregation tasks are more resilient to compression than retrieval.

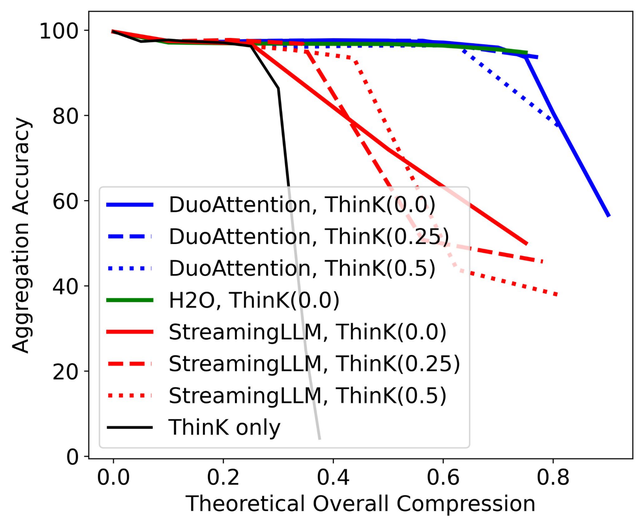


Figure 3: Aggregation Tasks Performance

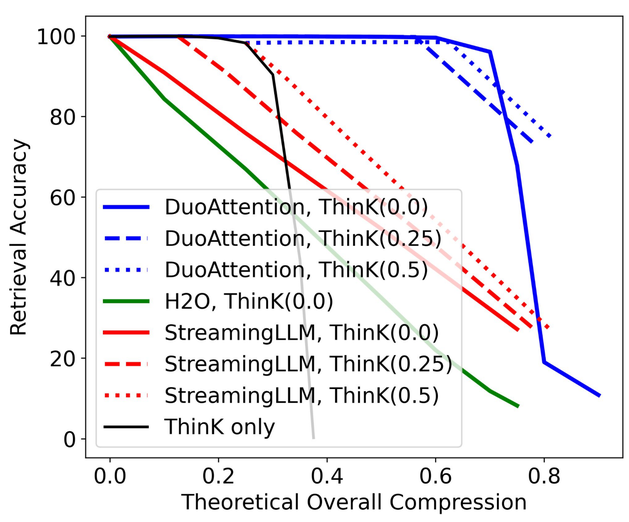


Figure 4: Retrieval Tasks Performance

Future Work

1. Do these results generalize across models, like Qwen3 or Gemma3?
2. Do these results generalize across benchmarks, like Long/InfiniteBench?
3. Does compression behavior remain consistent for longer context lengths?
4. What if we compose more compression methods (quantization/merging)?
5. How does composition perform on different tasks (QA/summarization)?
6. How does composition affect actual memory usage and inference speed?