CAOTE: Optimizing KV Cache Memory Through Attention Output Error-based Token Eviction

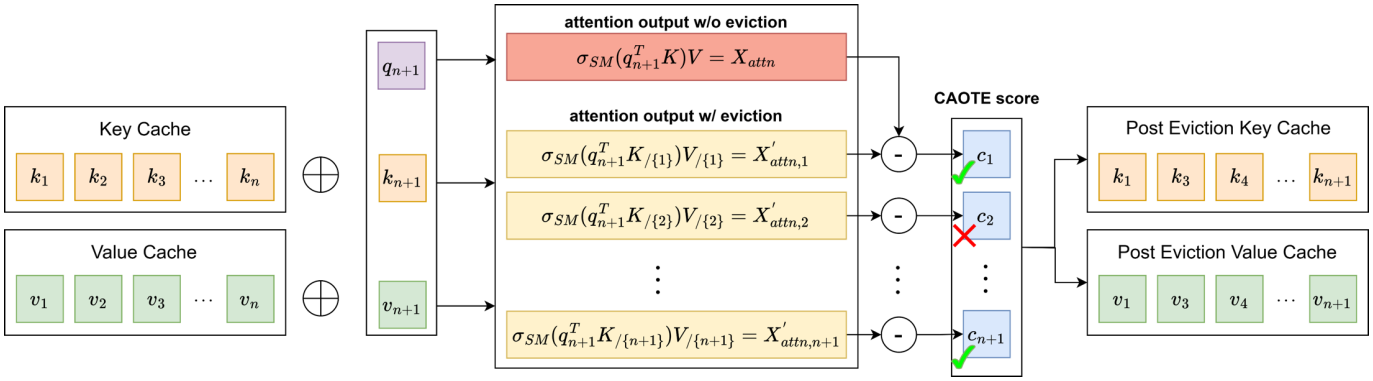
Qualcomm AI research

Raghavv Goel, Junyoung Park, Mukul Gagrani, Dalton Jones, Matthew Morse, Chris Lott, Harper Langston, Mingu Lee

{raghgoel, junpark, mgagrani, hlangsto, mingul}@qti.qualcomm.com

A token eviction rule which preserves attention fidelity by optimizing for

$$\text{argmin}_j\, e_{\text{eviction},j} = \left|\Delta X_{attn,j}\right|_2 = \left|X_{attn} - X'_{attn,j}\right|_2$$



Motivation

- **Problem**: KV cache dominates long-context memory
- **Gap**: Existing eviction ignores attention-output error and impact of value information
- **Idea**: Evict tokens that minimally change attention output

CAOTE

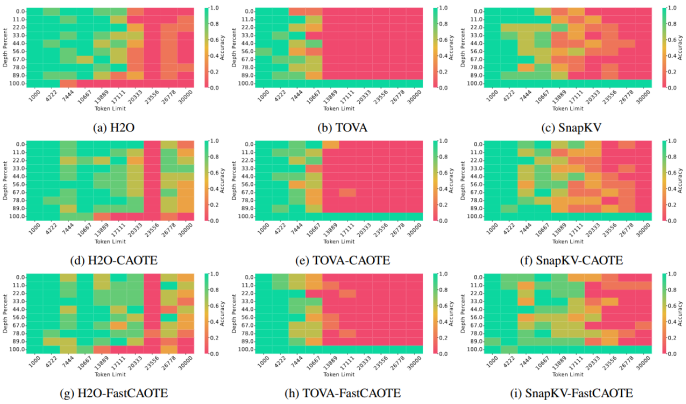
Evict tokens that minimally change attention output, not attention score

CAOTE score: $c_j = \left|\Delta X_{attn,j}\right|_2 = \frac{\alpha_j}{1-\alpha_j}\left|X_{attn} - v_j\right|_2$

Fast-CAOTE: efficient CAOTE

- Replace attention output with approximation
- **Reduce complexity**: $O(L^2 d) \rightarrow O(Ld)$
- Similar empirical performance to CAOTE.
- $c_j = \frac{\alpha_j}{1-\alpha_j}\left|\text{mean}(v_1, \ldots v_{b+1}) - v_j\right|_2$

LLama3.1-8B Needle in a haystack at 6k budget



Overall, CAOTE boosts the performance of all baselines, highlighting the importance of incorporating value information and minimizing attention-output deviation.

CAOTE as meta-token eviction

- CAOTE can be applied on **any existing** token-eviction methods.
- Convert scores of current methods by affine transformations to be b/w [0,1]
- Example: for H2O scores (h_j) are divided by sum of all scores.

$$c_j = \frac{h'_j}{1-h'_j}\left|X_{attn} - v_j\right|_2, \qquad h'_j = \frac{h_j}{\sum_{i=1}^{b+1} h_i}$$

Experiments

Llama3.1-8B LongBench at 2k budget

Method	Qasper	2WikiMQA	GovReport	TriviaQA	Avg (all)
Baseline	47.00	47.81	34.86	91.61	49.20
H2O	21.15	24.15	2.17	29.36	16.89
+Caote	38.34	**42.51**	15.11	63.60	33.31
+FastCaote	**41.27**	40.02	16.19	62.39	34.07
TOVA	37.26	34.48	21.17	90.73	37.52
+Caote	37.47	35.20	21.21	91.34	38.08
+FastCaote	38.22	36.93	21.72	**91.65**	38.19
SNAPKV	37.22	35.42	21.05	88.84	39.60
+Caote	37.49	37.26	21.67	90.65	<u>40.05</u>
+FastCaote	38.54	38.27	**21.97**	90.91	**40.73**

For QA tasks, H2O+CAOTE outperforms other baselines, even when vanilla H2O underperforms.

Perplexity on Wikitext

Budget	H2O			TOVA			SnapKV		
		+Caote	+FastCaote		+Caote	+FastCaote		+Caote	+FastCaote
Llama3.1-8B									
2k	2.007	1.884	1.891	-0.046	-0.088	-0.085	-0.019	-0.097	**-0.098**
4k	1.284	1.079	1.061	-0.047	-0.060	-0.058	-0.048	**-0.080**	-0.079
Llama3.2-3B									
2k	3.814	3.561	3.563	0.493	0.442	**0.432**	0.555	0.451	0.435
4k	2.460	2.142	2.128	0.175	0.150	**0.144**	0.223	0.152	**0.144**