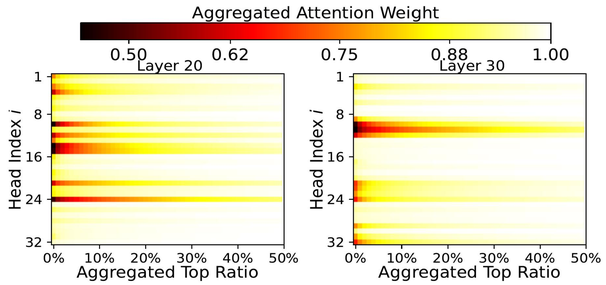
Ada-KV: Optimizing KV Cache Eviction by Adaptive Budget Allocation for Efficient LLM Inference

Yuan Feng, Junlin Lv, Yukun Cao, Xike Xie, S. Kevin Zhou

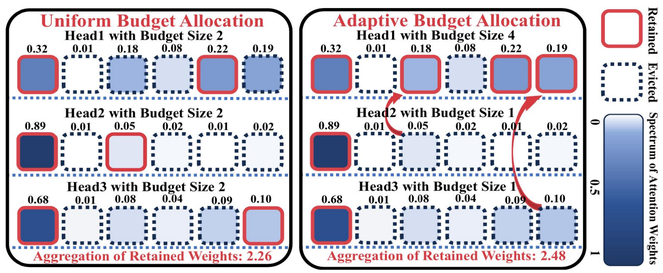
Abstract

Previous Works

The extensive KV cache required for long-sequence inference in LLMs presents significant efficiency challenges. Recent cache eviction methods aim to reduce the KV cache size but often **overlook the unique characteristics of different attention heads**, consequently allocating a uniform budget to each one.



The varying concentration characteristics of different attention heads necessitate an adaptive budget allocation strategy.



Our Ada-KV: From Uniform to Adaptive Allocation

- The first theoretical analysis to establish the necessity of head-wise adaptive allocation from a upper-bound pespective.
- The first head-wise adaptive budget allocation strategy capable of broadly enhancing existing cache eviction methods.

Theorical Anaylsis

Theorem 1: *The attention output loss introduced by cache eviction methods can be bounded by ε .*

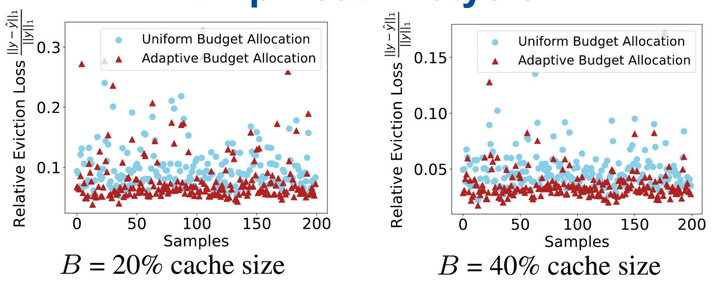
$$\epsilon = 2hC - 2C \sum_{i\in[1,h]} \sum_{j\in[1,n]} \mathcal{I}_i^j A_i^j$$

Theorem 2: *The adaptive budget allocation achieves the minimal loss upper bound ε* associated with cache eviction methods.*

$$\epsilon^* = \min_{\{\mathcal{I}_i\}} \epsilon = 2hC - 2C \sum_{i\in[1,h]} \sum_{A_i^j \in \textit{Top-k}(A_i, k=B_i)} A_i^j$$

Head-wise Adaptive Allocation minimize the Theoretical Loss Upper Bound.

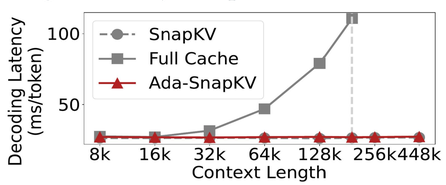
Empirical Analysis



The reduction in the theoretical loss upper bound translates to a significant decrease in the empirical attention output loss.

Efficient & Seamless Integration

Our adaptive strategy supports FlashAttention technique, achieving efficiency comparable to uniform allocation.



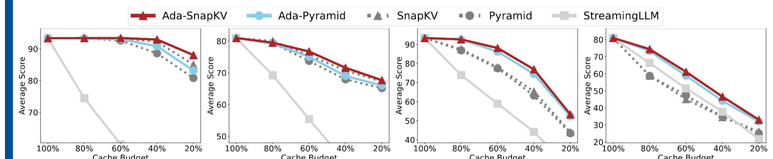
- **Official Integration: Part of the NVIDIA KVPRESS project.**
- **Community Adoption: Supported in 10+ public repositories.**

Evaluation

Integrated into SnapKV and PyramidKV

Evaluated on 29 datasets acorss Ruler and LongBench.

Two scenarios: **Question-aware and Question-agnostic**.



(a) Question-aware, Llama-3.1-8B-Instruct (b) Question-aware, Mistral-7B-Instruct-v0.2 (c) Question-agnostic, Llama-3.1-8B-Instruct (d) Question-agnostic, Mistral-7B-Instruct-v0.2

Figure 3: Average Score on Ruler Among 13 Datasets.

Table 2: Task Analysis for Llama-3.1-70B (Question-agnostic).

Domain	Full Cache	$b = 20\%$		$b = 40\%$	
		SnapKV	Ada-SnapKV	SnapKV	Ada-SnapKV
Single-Doc. QA	47.15	32.54	**36.05**	39.35	**42.21**
Multi-Doc. QA	60.07	46.98	**48.45**	54.07	**55.03**
Summarization	28.69	24.40	**24.81**	26.21	**26.57**
Few-shot	72.07	66.71	**67.80**	69.29	**70.29**
Synthetic	58.25	51.25	**51.50**	56.00	**56.75**
Code	47.82	**52.44**	51.51	**51.90**	49.23
Ave.	52.25	44.95	**46.08**	48.91	**49.64**

Findings

- Question-agnostic scenarios—such as multi-turn dialogues—pose greater challenges, warranting more attention in future.
- Ada-KV demonstrates broad applicability, improving existing methods under various tasks, budgets and scenarios.

Ada-KV has been widely adopted in many follow-up works, garnering over 60 citations.

Table 3: Performance gains from applying the Ada-KV strategy to follow-up methods. All results are from the Llama-3.1-8B model on the Longbench benchmark under question-agnostic settings.

Cache	DuoAttn *Training-based*	HeadKV *Training-based*	CriticalKV w/o. Ada-KV	CriticalKV w/. Ada-KV	DefensiveKV w/o. Ada-KV	DefensiveKV w/. Ada-KV
20%	39.52	42.64	42.99	43.77	43.78	**46.68**
40%	48.17	47.23	47.29	48.00	47.76	**49.21**