# ArkVale: Efficient Generative LLM Inference with Recallable Key-Value Eviction

Renze Chen[1], Zhuofeng Wang[1], Beiquan Cao[1], Tong Wu[1], Size Zheng[1], Xiuhong Li[1], Xuechao Wei[1], Shengen Yan[2], Meng Li[1], Yun Liang[1]

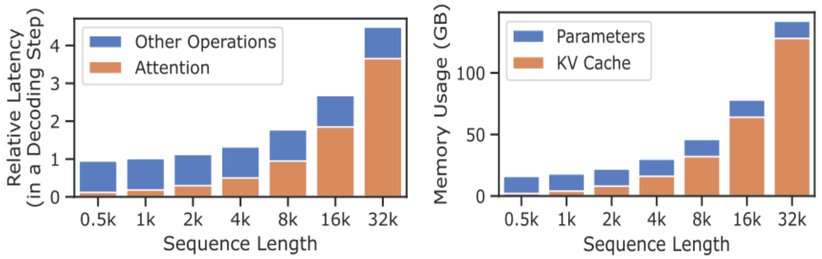
[1]Peking University [2]Infinigence-AI E-mail: crz@pku.edu.cn Code: https://github.com/pku-liang/ArkVale

Scan to access our code

## 1. Motivation

Impact of Long-context

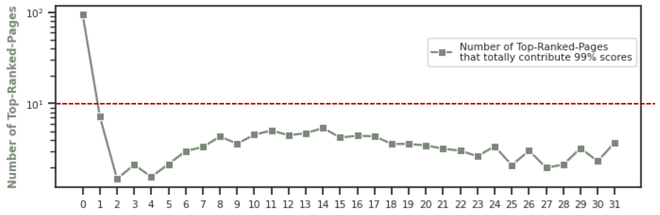
- Long context attention can be the bottleneck of LLM decoding
- Long context can be the memory bottleneck of LLM decoding, which hampers the use of larger batch-size for serving.




LongChat-7b-v1.5-32k (batch-size=8)

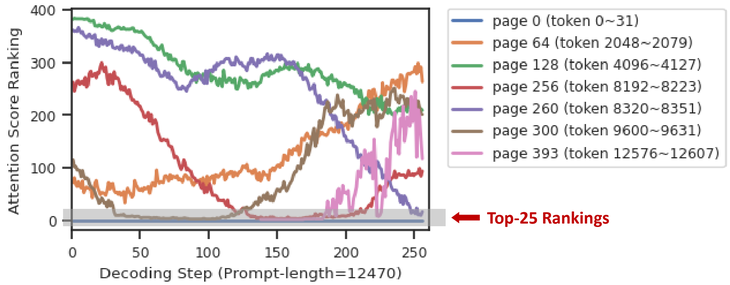
## 2. Observation

In most layers, less than 10 KV-cache pages contributing most attention scores.



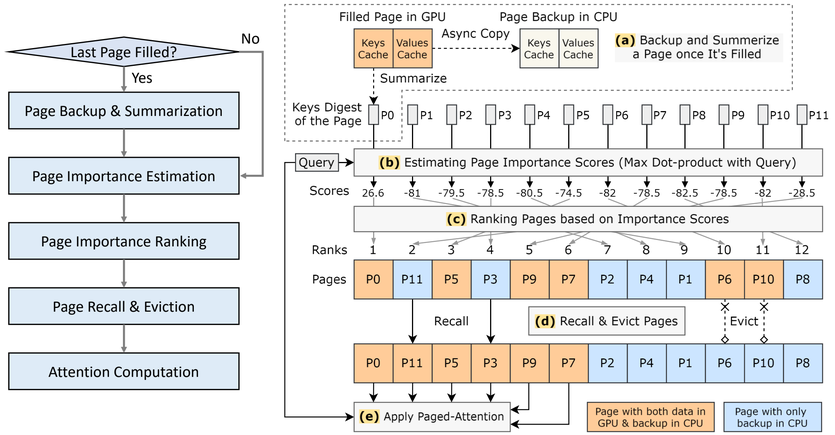

LongChat-7b-v1.5-32k page-size=32

Importance of KV-cache token/page can dynamically change overtime

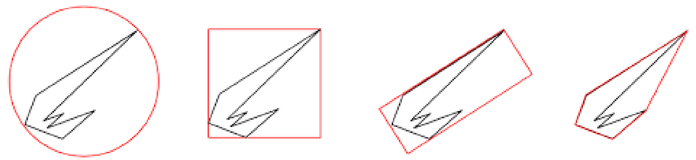


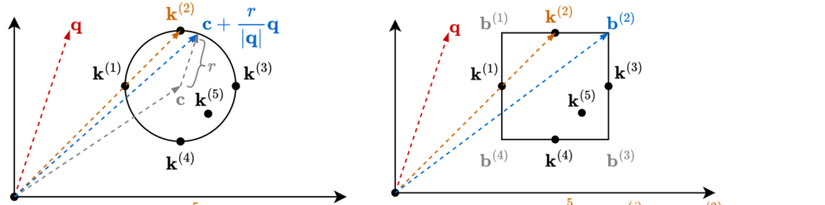

## 3. Techniques

Workflow Overview




Page Summarization & Importance Estimation using Bounding-volume



$$\text{Page-Importance} = \max_{i=1}^{5} \mathbf{q}\cdot\mathbf{k}^{(i)} = \mathbf{q}\cdot\mathbf{k}^{(2)} \approx \mathbf{q}\cdot(\mathbf{c}+\frac{r}{|\mathbf{q}|}\mathbf{q}) = \mathbf{q}\cdot\mathbf{c}+r|\mathbf{q}|$$

(a) Bounding-sphere need to record the center $\mathbf{c}$ and the radius $r$.

$$\text{Page-Importance} = \max_{i=1}^{5} \mathbf{q}\cdot\mathbf{k}^{(i)} = \mathbf{q}\cdot\mathbf{k}^{(2)} \approx \max_{i=1}^{4} \mathbf{q}\cdot\mathbf{b}^{(i)} = \mathbf{q}\cdot\mathbf{b}^{(2)} = \mathbf{1}\cdot\max(\mathbf{q}\odot\mathbf{b}^{(2)}, \mathbf{q}\odot\mathbf{b}^{(4)})$$

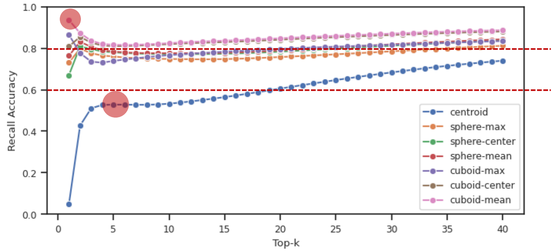
(b) Bounding-cuboid need to record the max-vector $\mathbf{b}^{(2)} = \max_{i=1}^{5}\mathbf{k}^{(i)}$ and the min-vector $\mathbf{b}^{(4)} = \min_{i=1}^{5}\mathbf{k}^{(i)}$.

Figure 6: Summarize page keys $\{\mathbf{k}^{(i)}\}_{i=1}^{5}$ into their bounding-volume (sphere/cuboid). We can estimate the max-dot-product between query $\mathbf{q}$ and keys $\{\mathbf{k}^{(i)}\}_{i=1}^{5}$ using the bounding-volume.

## 4. Evaluation

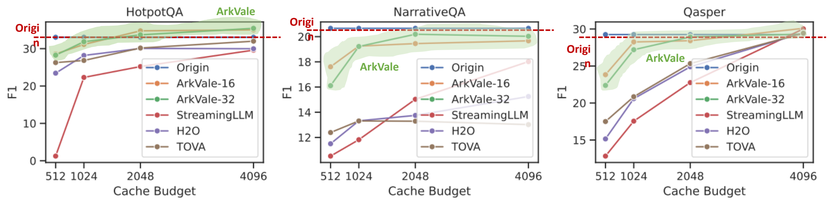
Top-k recall accuracy of different importance estimation methods

- Baseline method (centroid) cannot achieve even 60% top-5 recall accuracy.
- Our cuboid-mean method ensure 95% top-1 recall accuracy, and can achieve 80% top-$k$ recall accuracy for all $k$.



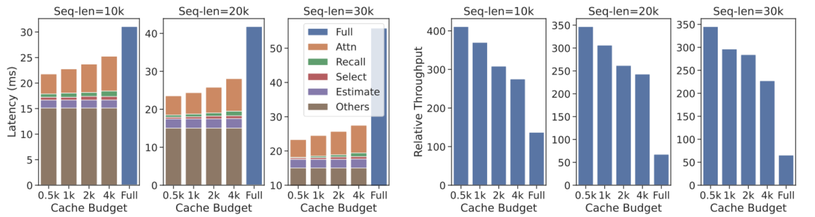

Part of Evaluation Results on Long-Bench

- ArkVale can surpass all baselines with different datasets and cache-budgets.
- ArkVale can approach or even surpass "Origin".
- ArkVale-16 (page-size=16) usually outperforms ArkVale-32 (page-size=32).




Decoding Latency & Throughput Evaluation

- Allocate 40 GB GPU memory for KV-cache (and page digests) on A100 GPU.
- Compared to baseline, ArkVale can achieve up to 2.2x decoding speedup.
- Compared to baseline, ArkVale can achieve up to 6x decoding throughput.




(a) Latency Breakdown (batch-size=4)

(b) Throughput Comparison