MIT HAN LAB

LServe: Efficient Long-sequence LLM Serving with Unified Sparse Attention

Shang Yang*, Junxian Guo*, Haotian Tang, Qinghao Hu,
Guangxuan Xiao, Jiaming Tang, Yujun Lin, Zhijian Liu, Yao Lu, Song Han

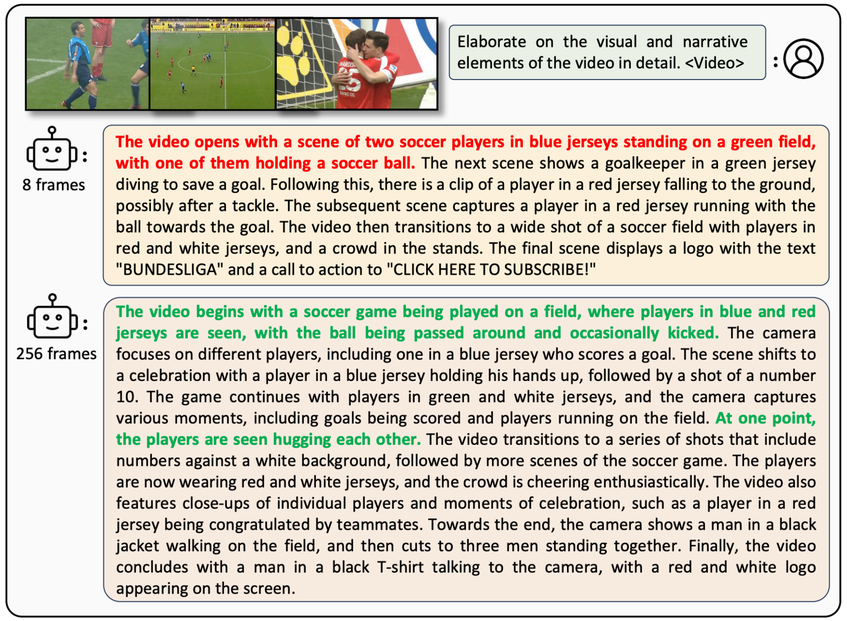
MIT, NVIDIA, SJTU. Accepted by MLSys 2025.

* indicates the equal contribution to this work.

Long-sequence LLM serving is crucial

LServe optimizes prefilling & decoding speed and memory consumption

- LLM needs to process long sequences for applications like long document/video understanding
- Reasoning models demonstrate impressive capabilities with inference time scaling, solving complex tasks with long decoding trajectories extending to over **20k tokens**.



a 224×224 image = 256 tokens

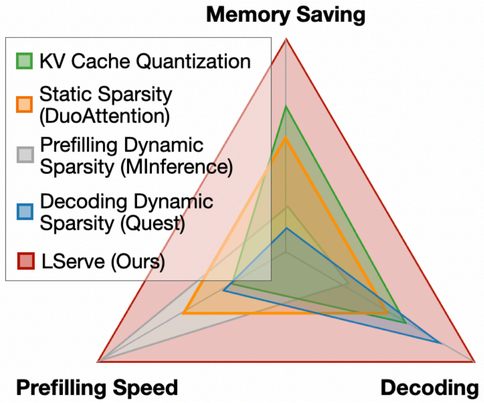
a half-hour video at 1 FPS = **0.5M tokens**

Reasoning Model:

Decode 20K tokens ~ Prefill 512K tokens

TensorRT-LLM	Prefilling (0.5M tokens)	Decoding (20K tokens)
Latency	454 seconds	540 seconds

LServe: Efficient Long-Sequence LLM Serving



Sparse Attention Unification

- Unified Framework for Sparse Attention
- Hybrid Static & Dynamic Sparsity

System-Algorithm Co-optimization

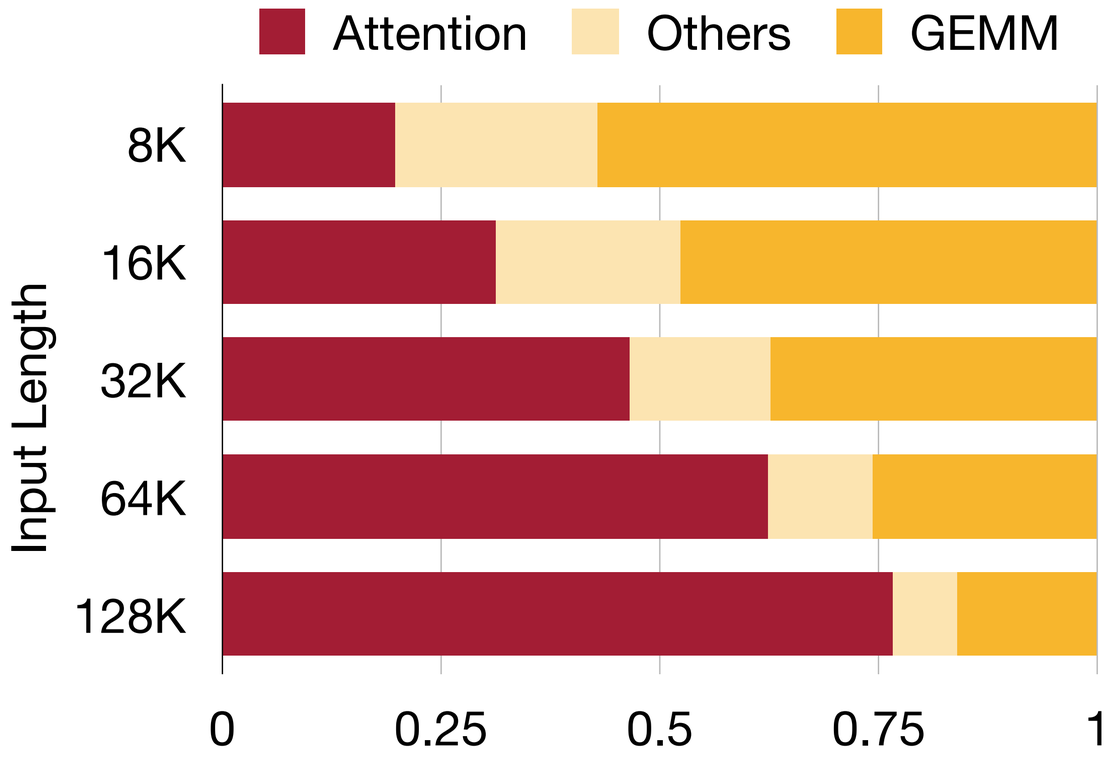
- Hierarchical Paging System
- Reusable Page Selector

O1 Replication Journey: A Strategic Progress Report [Qin *et al.*, arXiv 2024]

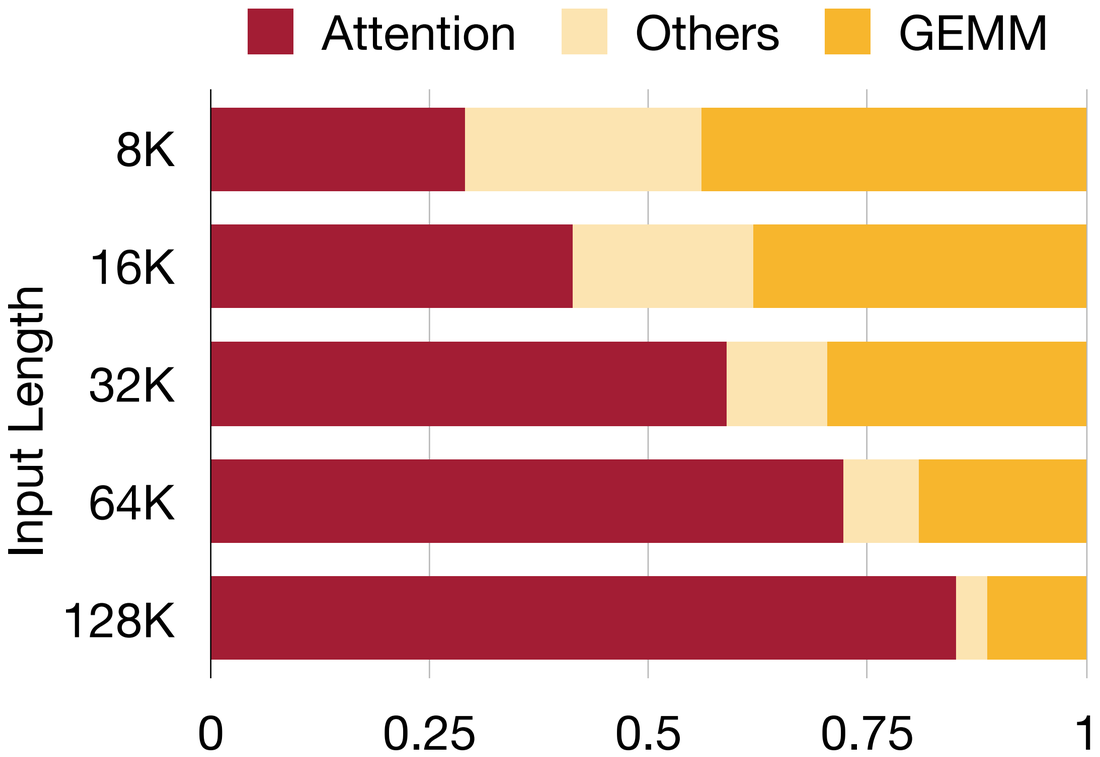
LongVILA: Scaling Long-Context Visual Language Models for Long Videos [Chen *et al.*, ICLR 2025]

Attention is important

Attention becomes LLM serving bottleneck as sequence gets longer



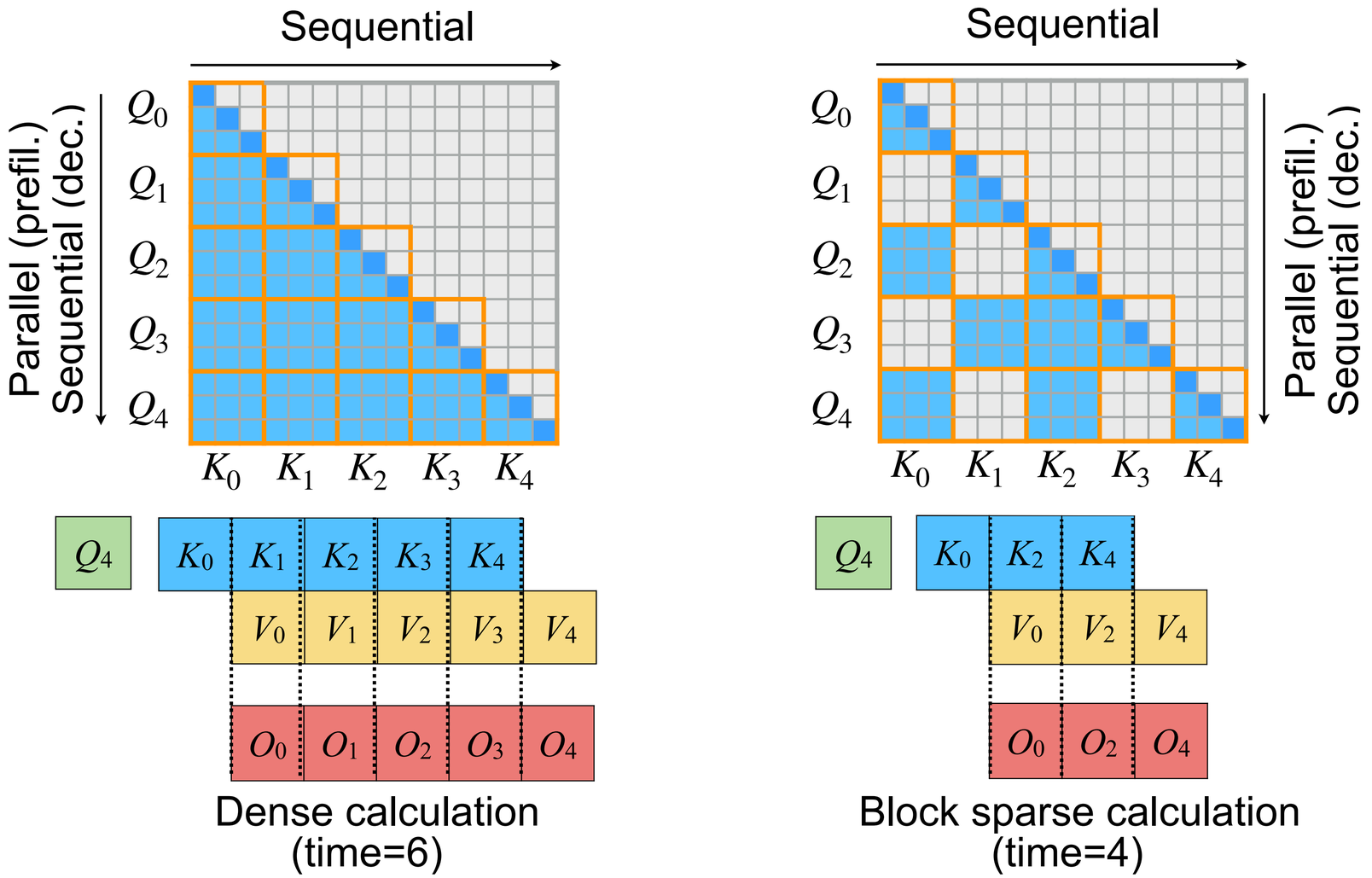
(a) Latency breakdown of LLM prefilling



(b) Latency breakdown of LLM decoding

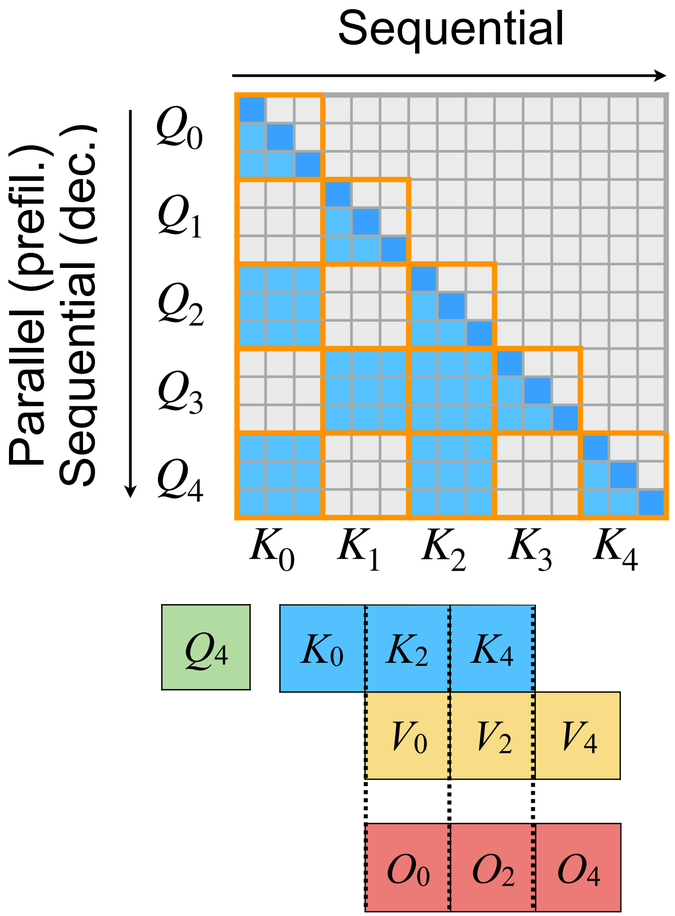
Sparse Attention on GPUs

How is attention calculated on GPU

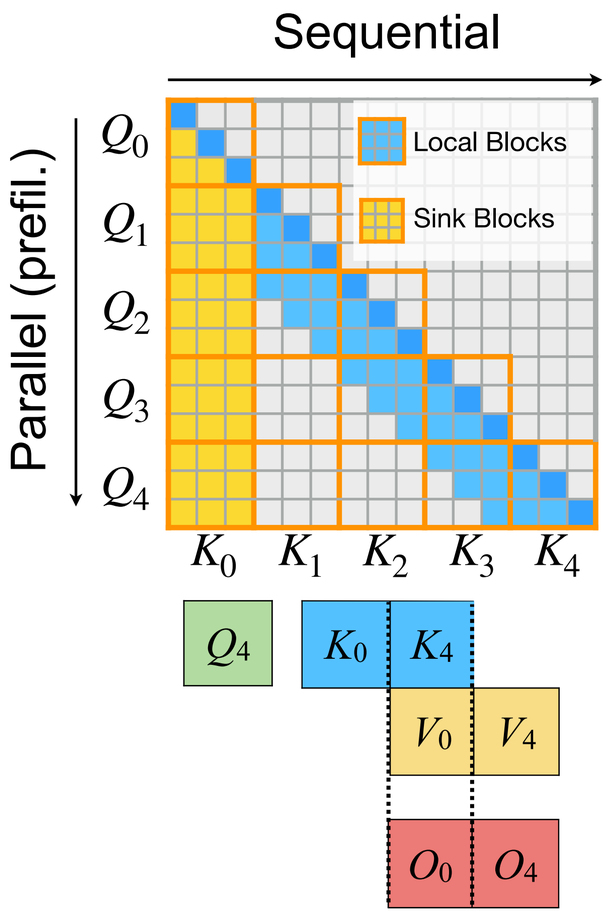


Sparse Attention on GPUs

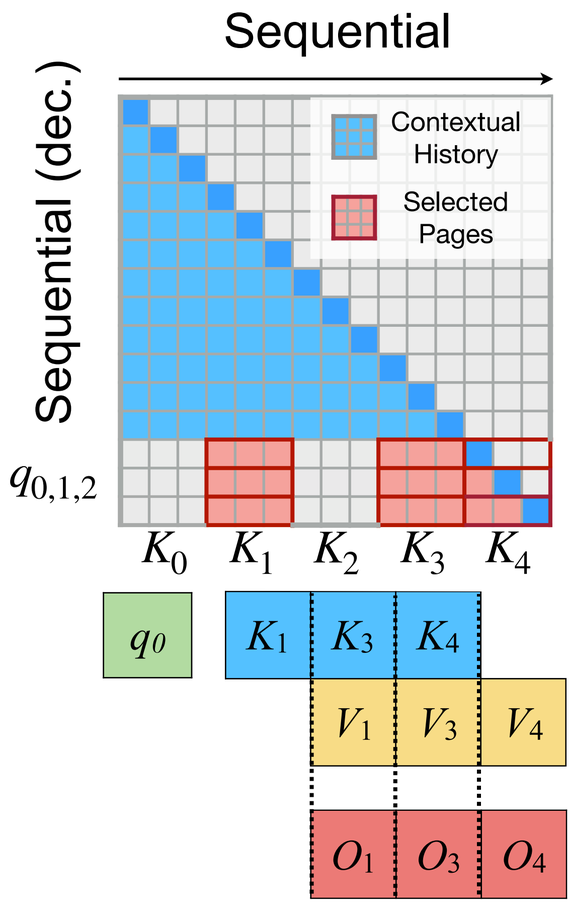
Unified Hybrid Sparse Attention in LServe



(a) Block Sparse Attention



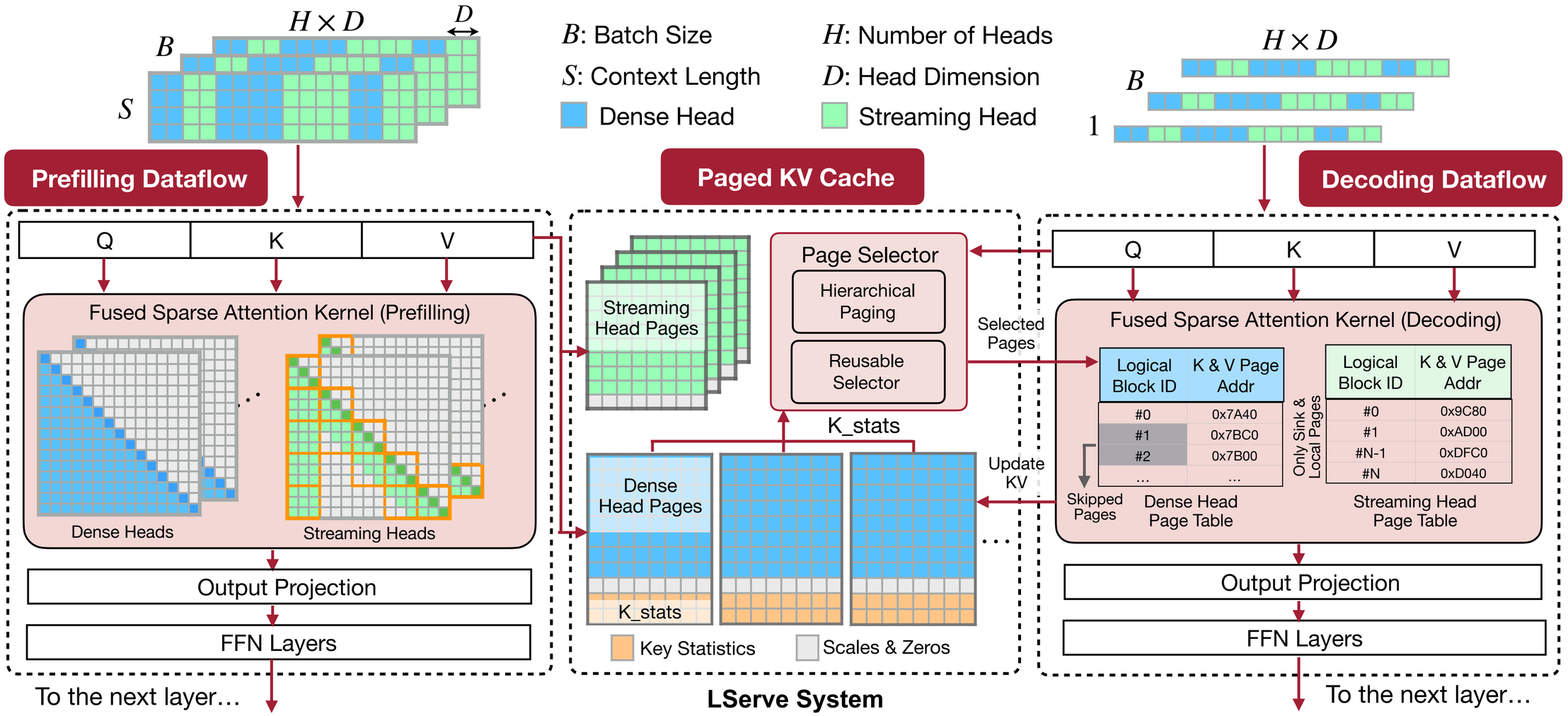
(b) Block-Sparse: Streaming



(c) Block-Sparse: Page Pruning

LServe System Overview

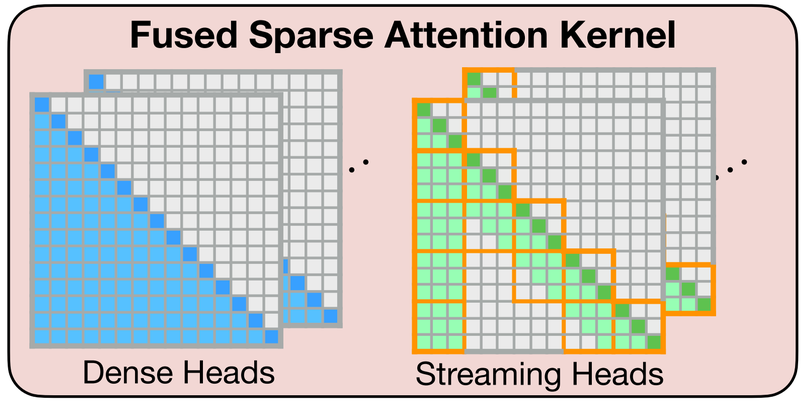
LServe accelerates both profiling and decoding stage w/ Unified Sparse Attention

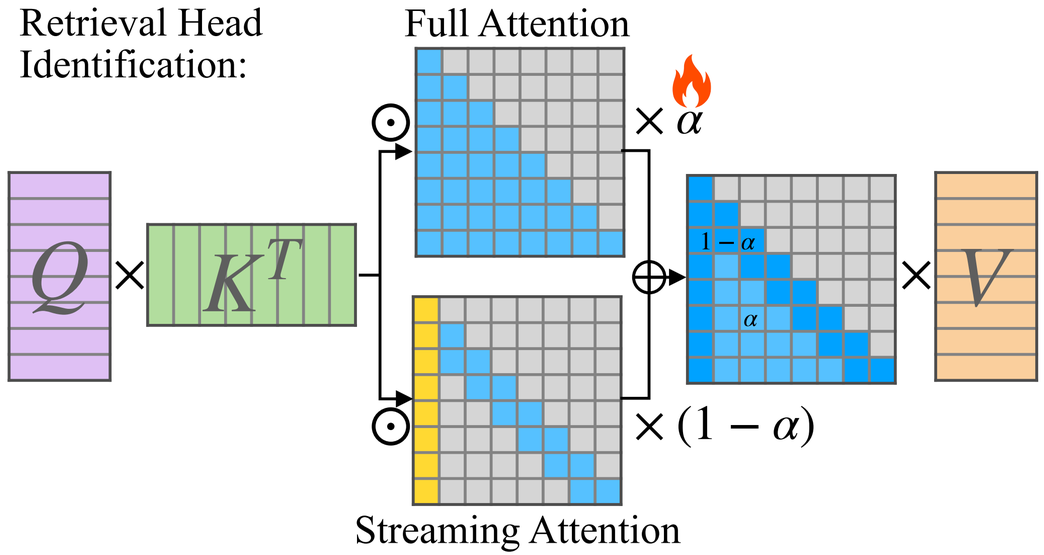


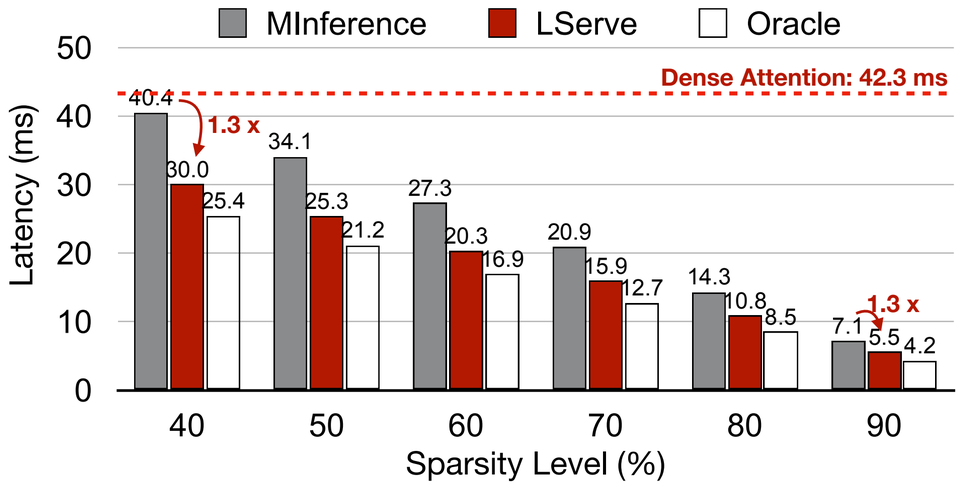
Prefilling Stage in LServe

Static sparsity with fused sparse attention template

- Inspired by DuoAttention, we instantiate the fused attention kernel for *static sparsity*
- Conditional statements and branching operations can be expensive on GPUs
- We introduce **iterator-based** memory loading abstraction for streaming heads





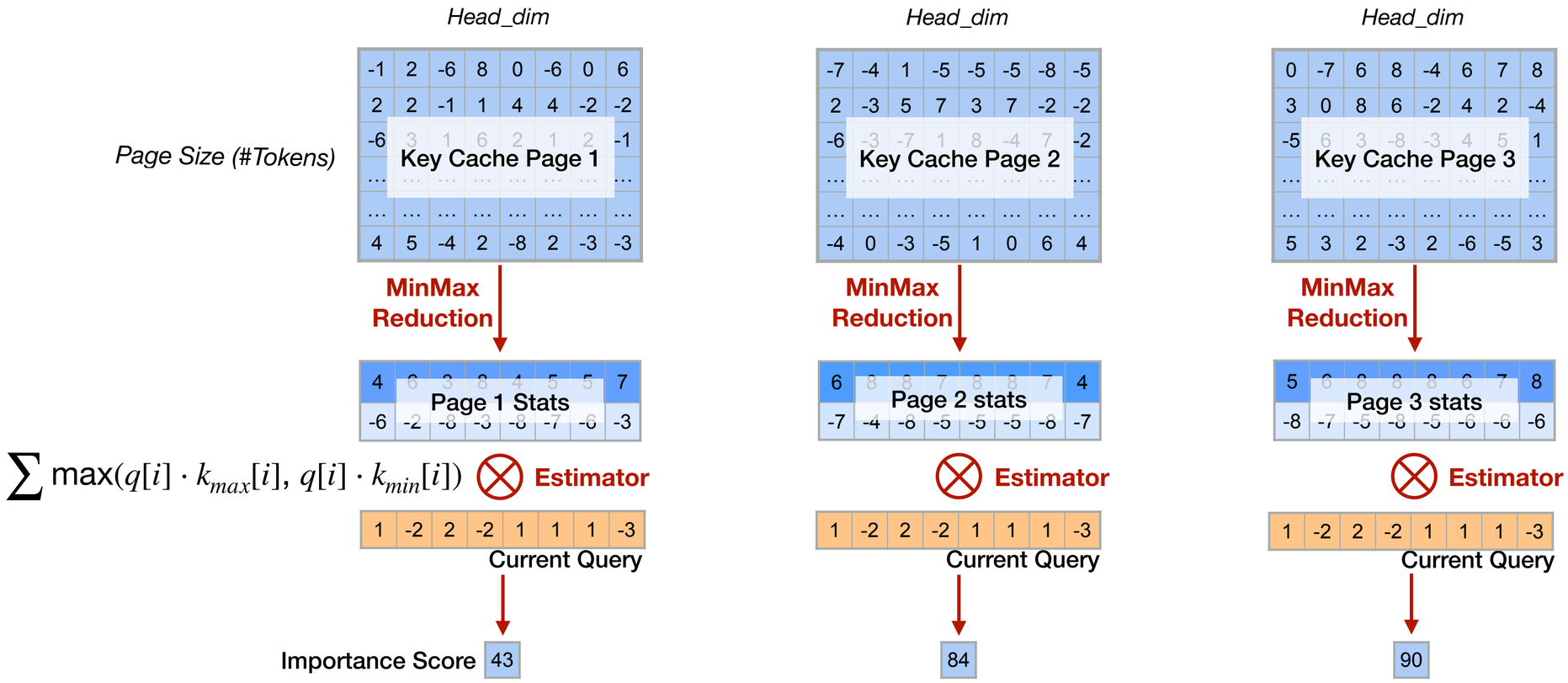


DuoAttention: Efficient Long-Context LLM Inference with Retrieval and Streaming Heads [Xiao et al., ICLR 2025]

Dynamic Sparsity in Decoding Stage

Vanilla Pruning with Page Importance Estimation

- Query-aware selective attention:
  - Statistical representatives for page importance estimation

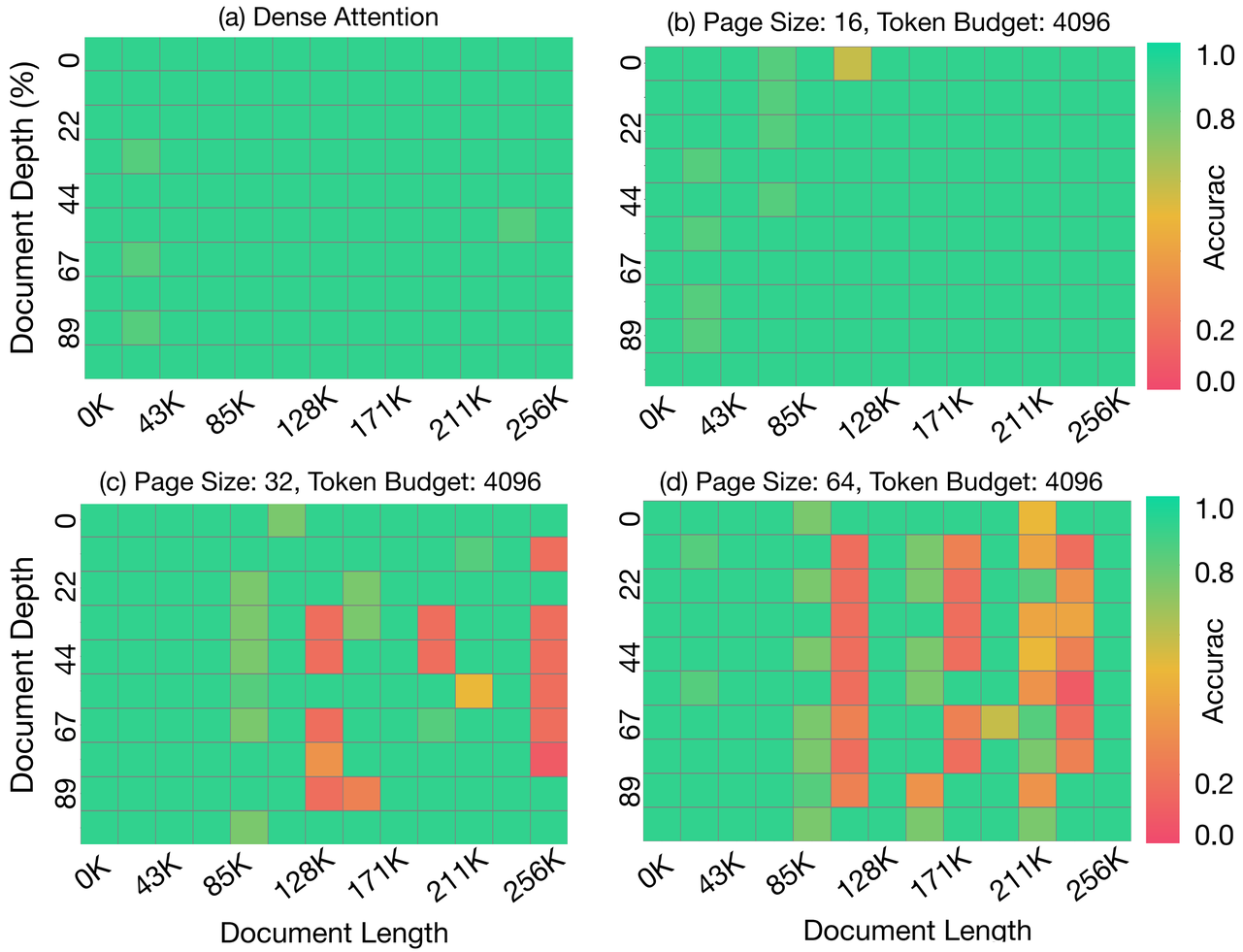


Quest: Query-Aware Sparsity for Efficient Long-Context LLM Inference [Tang et al., ICML 2024]

Hierarchical Paging System

Page size dilemma: Accuracy-efficiency tradeoff

- Increasing page size may lead to higher throughputs but lower accuracy with sparse attention



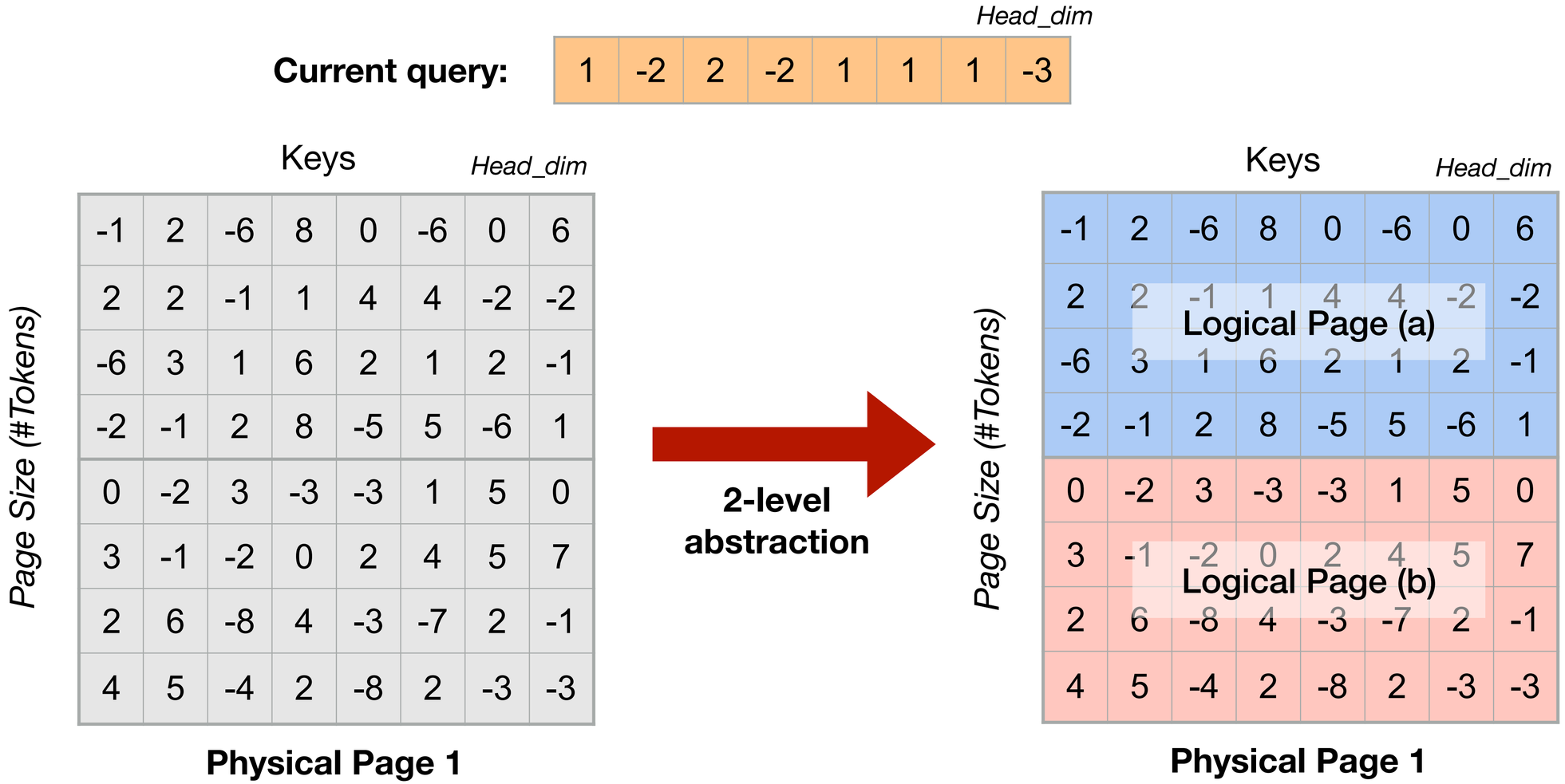
Seq_len	Page Size			
	16	32	**64**	128
512	11.0 ms	10.7 ms	**10.5 ms**	10.5 ms
1024	13.8 ms	13.0 ms	**12.7 ms**	12.7 ms
2048	22.1 ms	20.1 ms	**18.3 ms**	18.2 ms
4096	35.7 ms	31.6 ms	**28.1 ms**	28.1 ms
8192	**77.1 ms**	**63.0 ms**	**51.0 ms**	50.6 ms
Max Slowdown	1.52×	1.25×	1.01×	1.00×

per-step decoding latency w/ different page sizes

Hierarchical Paging System

Hierarchical Paging: Decouple physical layout & pruning algorithm

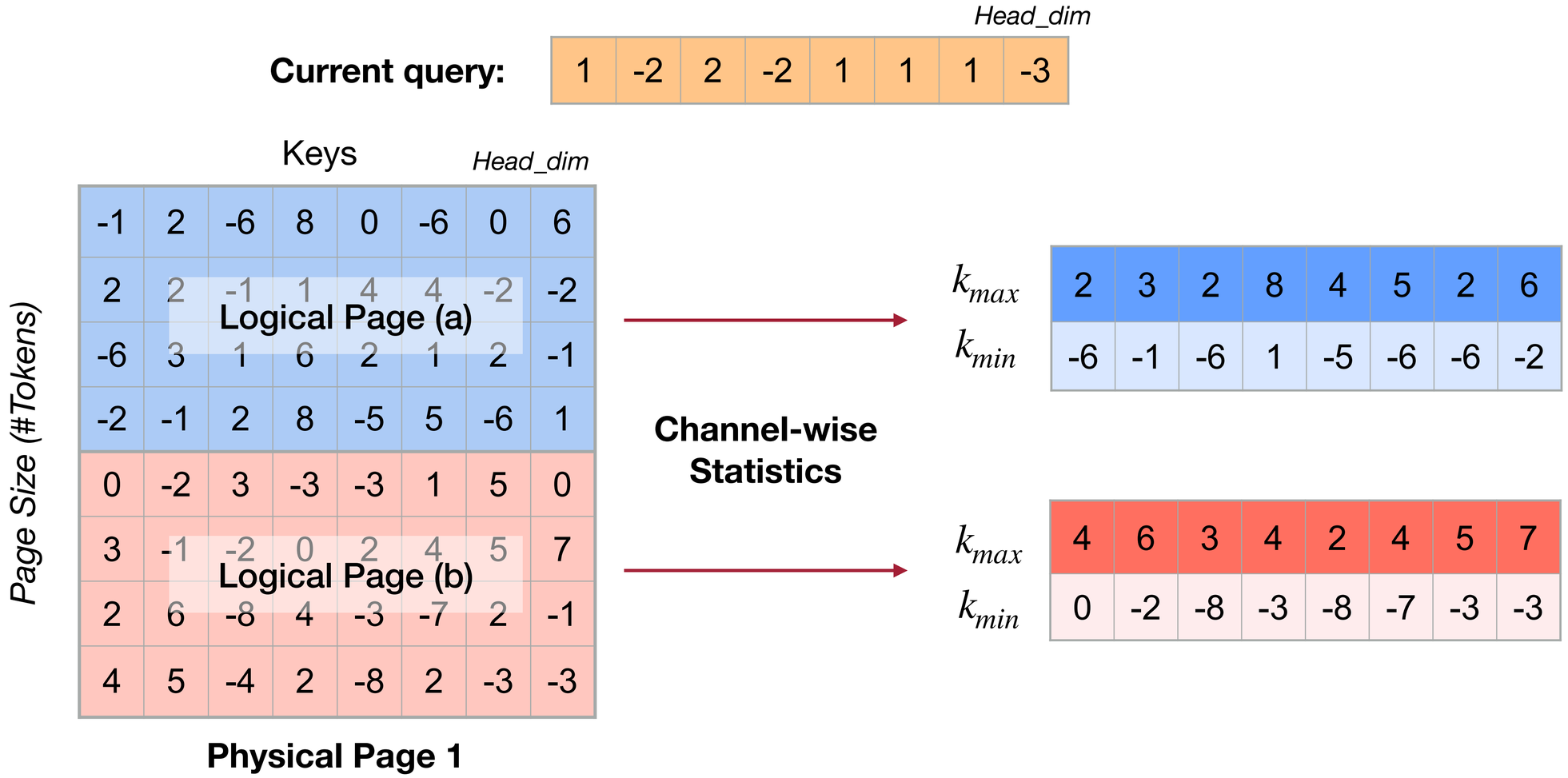
- Instead of setting one significance indicator per-page, we introduce 2-level selection algorithm



Hierarchical Paging System

Hierarchical Paging: Decouple physical layout & pruning algorithm

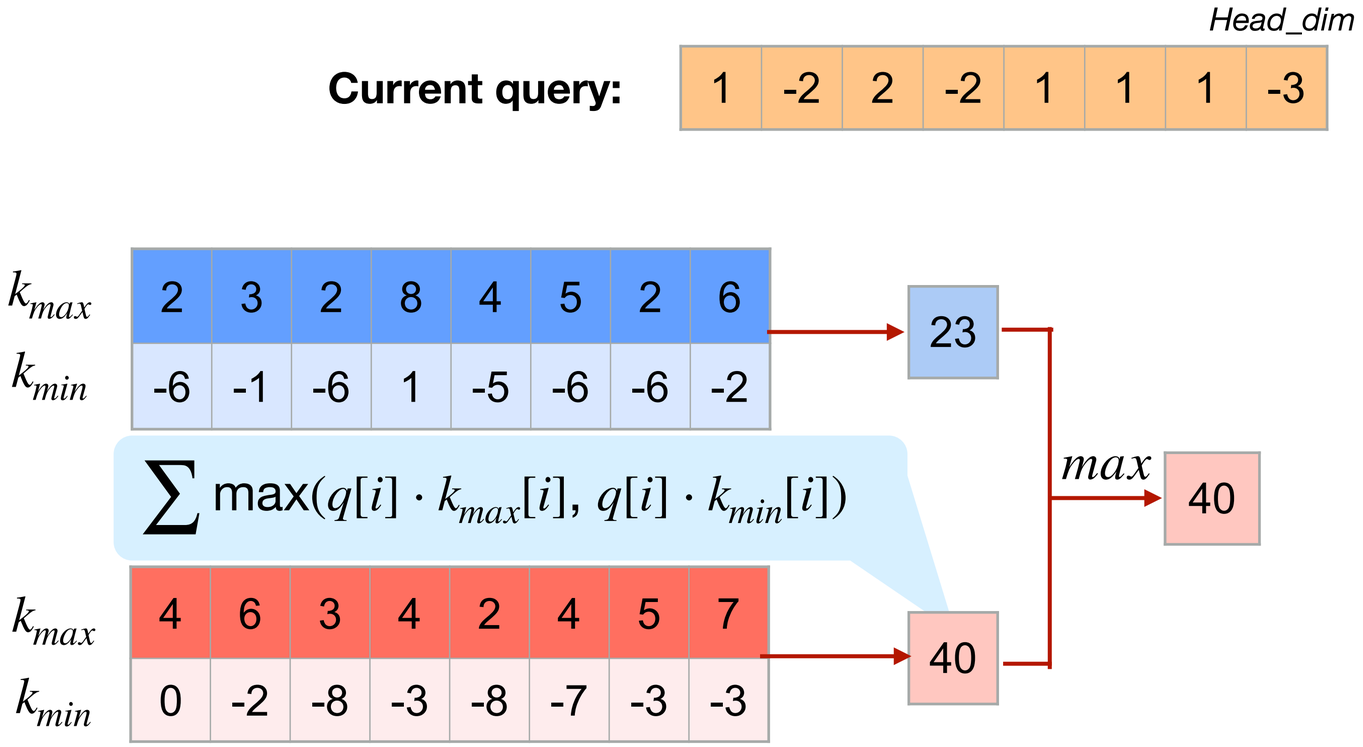
- Instead of setting one significance indicator per-page, we introduce 2-level selection algorithm



Hierarchical Paging System

Hierarchical Paging: Decouple physical layout & pruning algorithm

- Instead of setting one significance indicator per-page, we introduce 2-level selection algorithm



$$\sum \max(q[i] \cdot k_{max}[i], q[i] \cdot k_{min}[i])$$

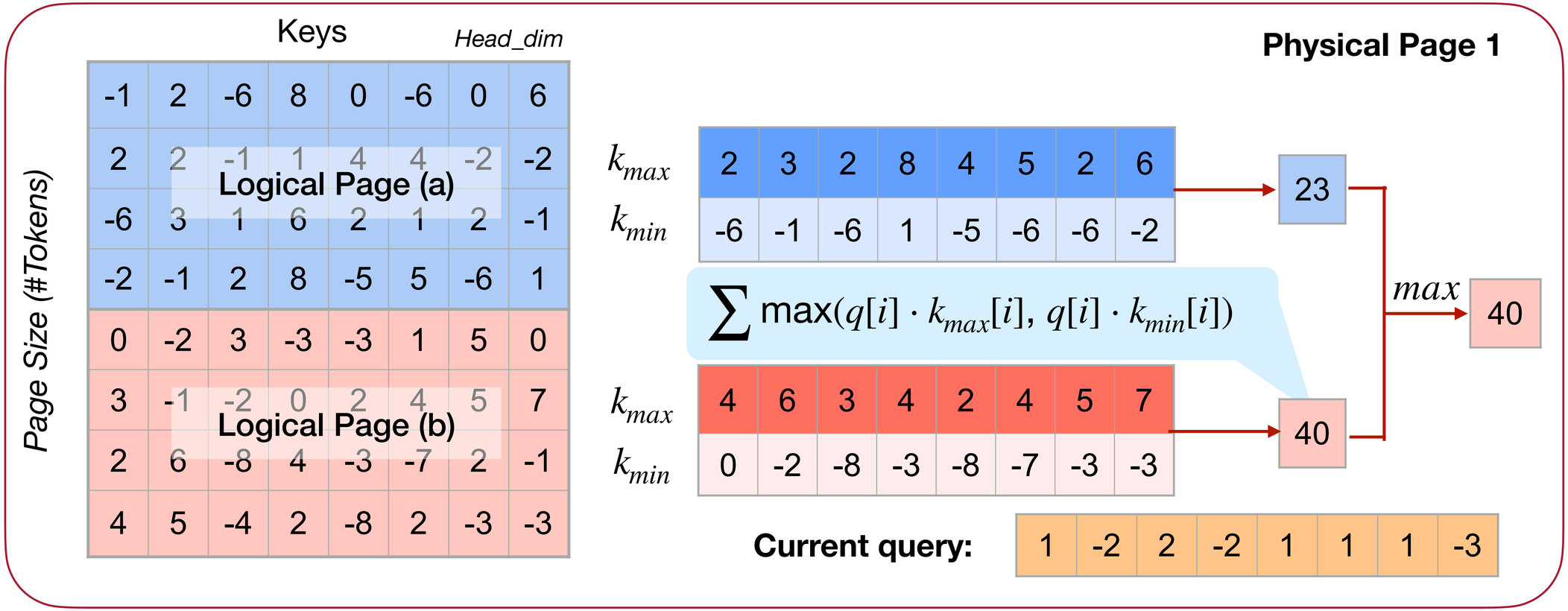
$$= 1 \cdot 2 + (-2) \cdot (-1) + 2 \cdot 2 + (-2) \cdot 1 + 1 \cdot 4 + 1 \cdot 5 + 1 \cdot 2 + (-3) \cdot (-2)$$

$$= 23$$

Hierarchical Paging System

Hierarchical Paging: Decouple physical layout & pruning algorithm

- Instead of setting one significance indicator per-page, we introduce 2-level selection algorithm



-1	2	-6	8	0	-6	0	6
2	2	-1	1	4	4	-2	-2
-6	3	1	6	2	1	2	-1
-2	-1	2	8	-5	5	-6	1
0	-2	3	-3	-3	1	5	0
3	-1	-2	0	2	4	5	7
2	6	-8	4	-3	-7	2	-1
4	5	-4	2	-8	2	-3	-3

k_{max}	2	3	2	8	4	5	2	6
k_{min}	-6	-1	-6	1	-5	-6	-6	-2

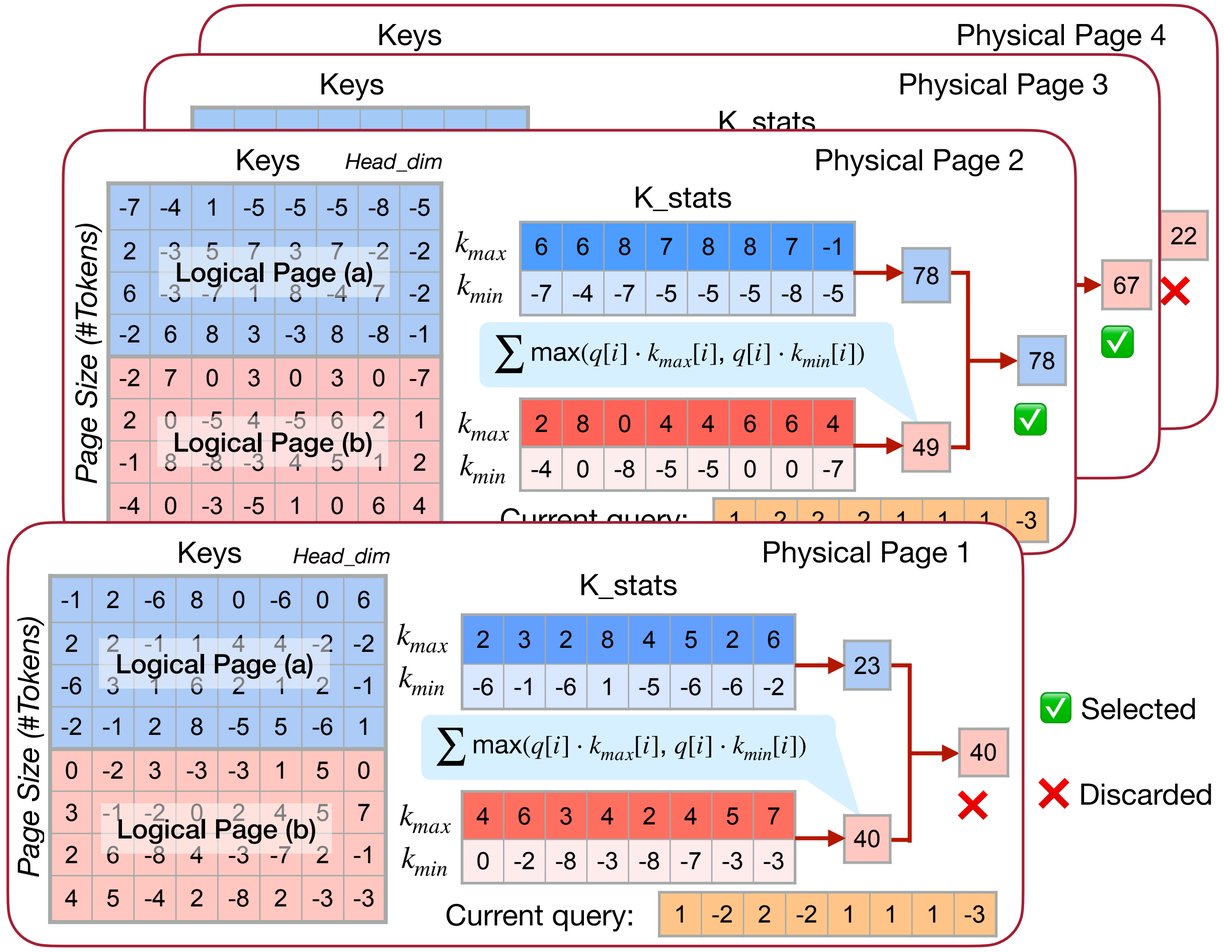
k_{max}	4	6	3	4	2	4	5	7
k_{min}	0	-2	-8	-3	-8	-7	-3	-3

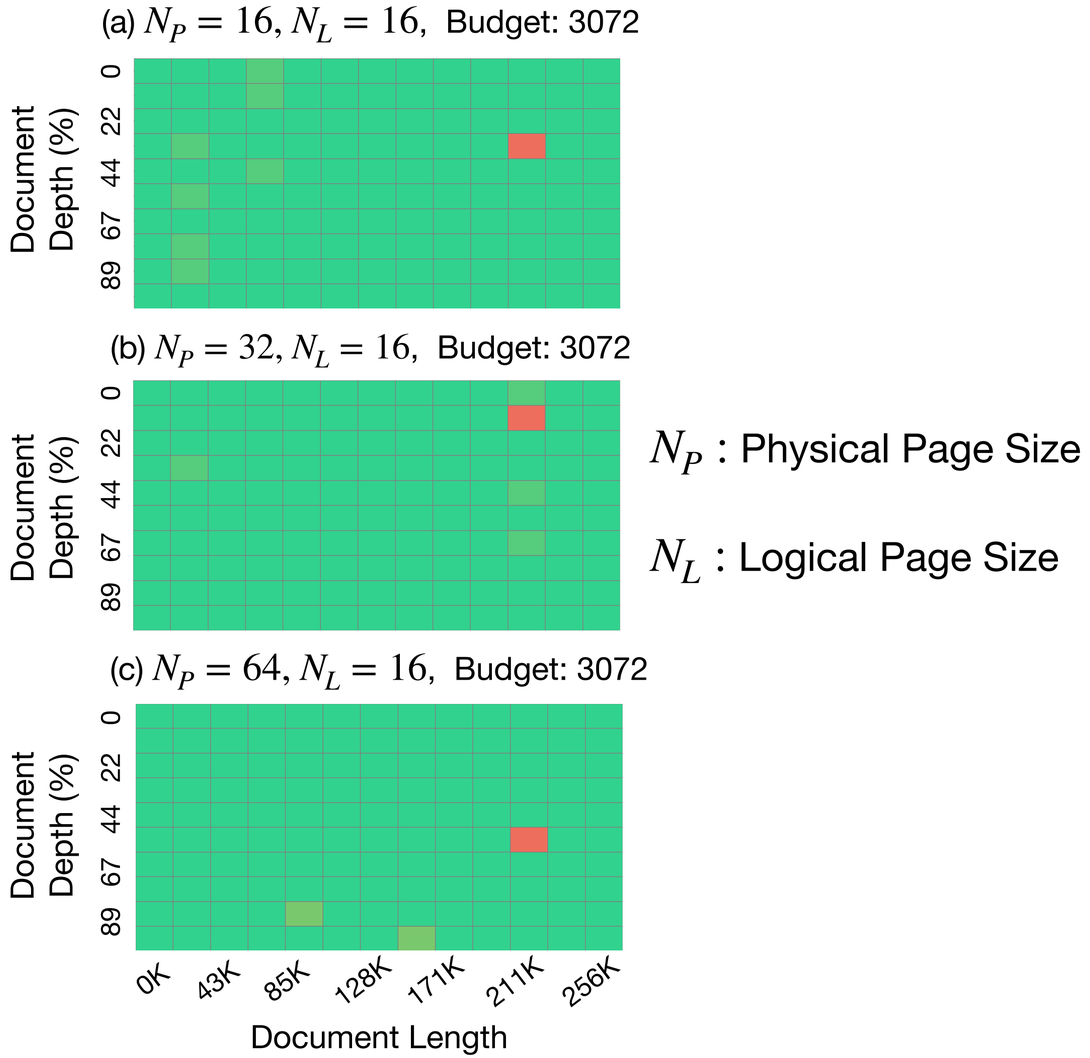
Current query:	1	-2	2	-2	1	1	1	-3

Hierarchical Paging System

Page size dilemma: Accuracy-efficiency tradeoff

- Increasing page size may lead to higher throughputs but lower accuracy with sparse attention

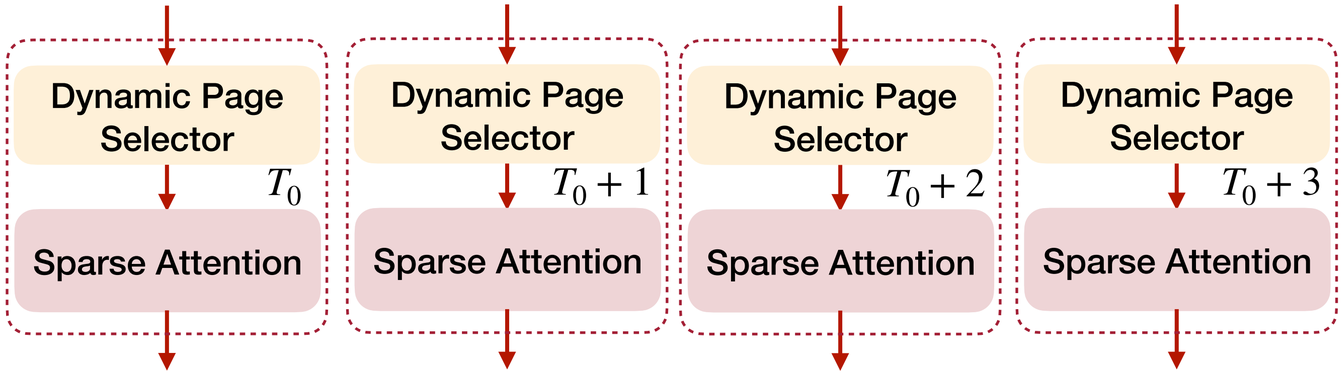




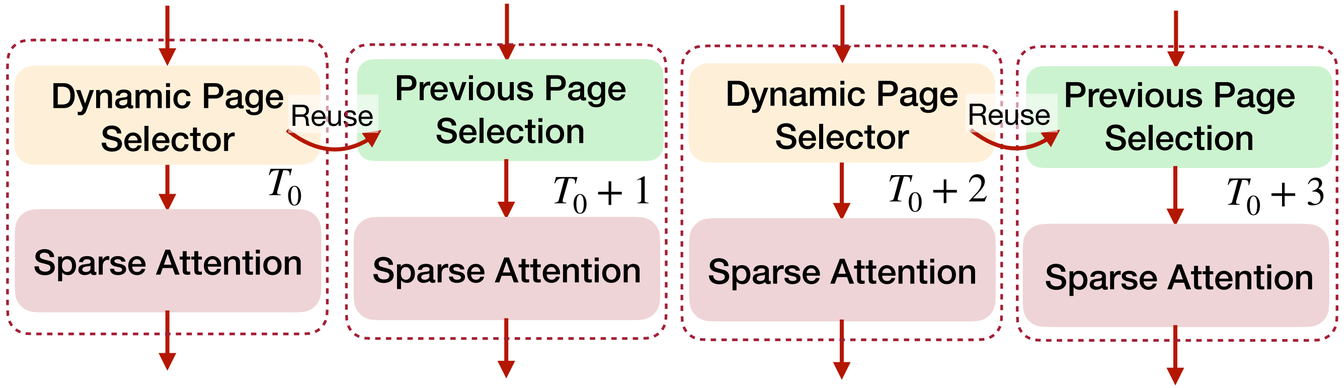
Reusable Page Selector

Reduce the overhead of page selection with temporal locality

- Query-aware page selector can be the bottleneck of LLM inference
- The effectiveness of hierarchical paging system indicates the **spatial locality** of natural language
- With **temporal locality**, consecutive query tokens can reuse the selected KV pages



(a) Vanilla pipeline of dynamic sparse attention

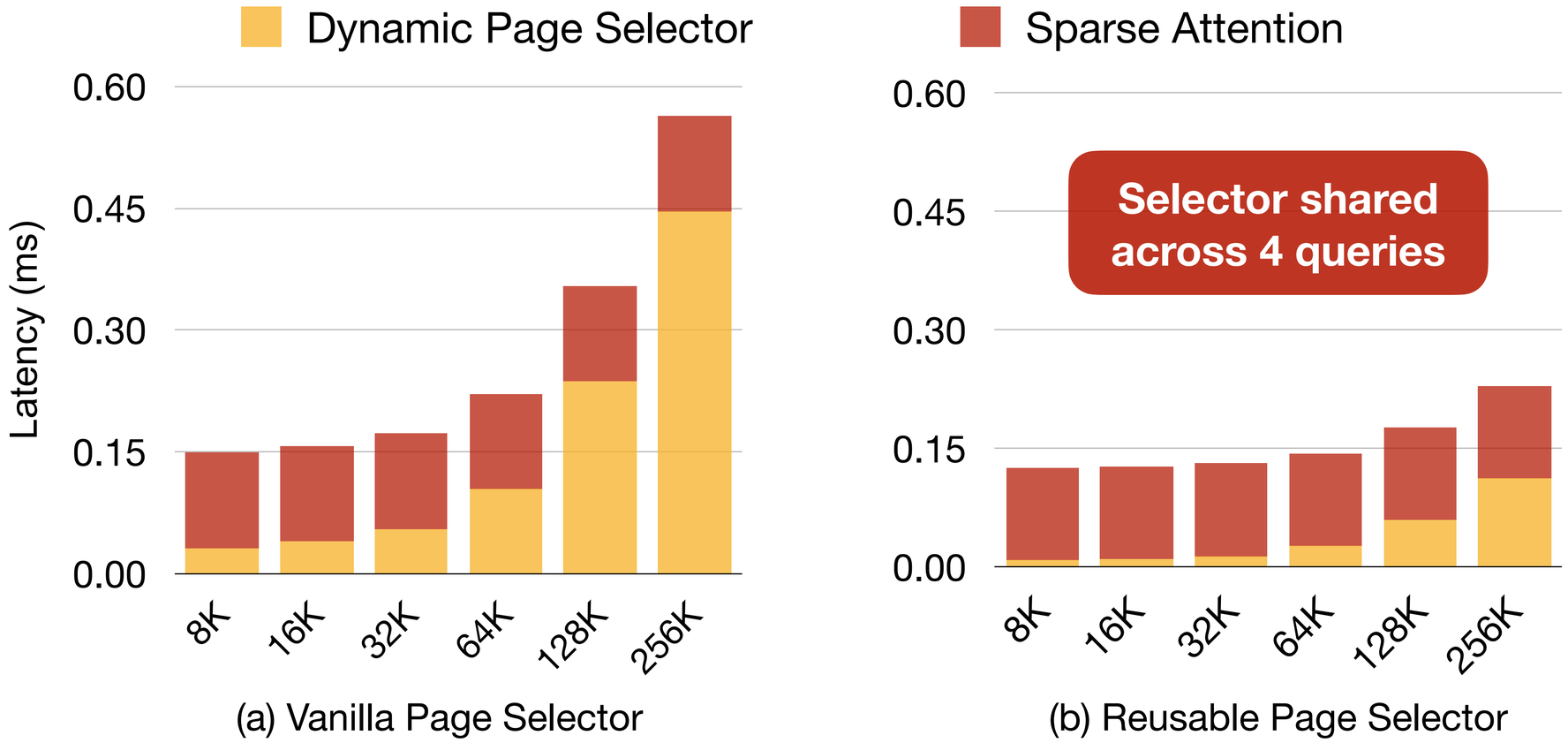


(b) Sparse attention with reusable page selector

Reusable Page Selector

Reduce the overhead of page selection with temporal locality

- The effectiveness of hierarchical paging system indicates the **spatial locality** of natural language
- With **temporal locality**, consecutive query tokens can reuse the selected KV pages

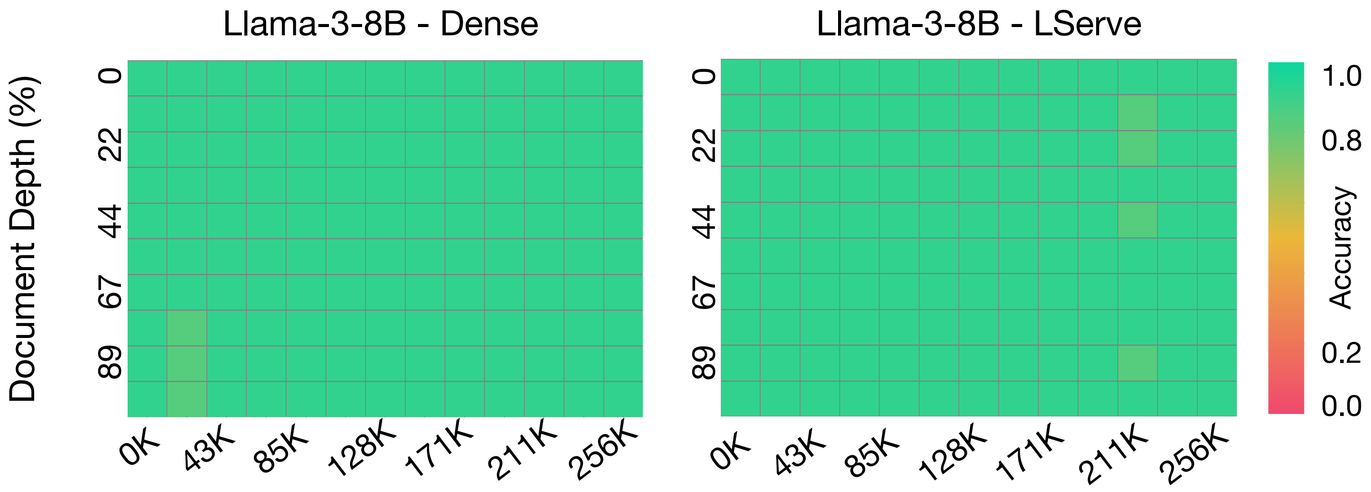


(a) Vanilla Page Selector

(b) Reusable Page Selector

Accuracy Evaluation

LServe preserves models' accuracy across long-context benchmarks



(a) LServe on Needle-In-A-Haystack benchmark

Model	DS-R1-Llama-8B	
Benchmark	Dense	LServe
AIME@2024	43.3	43.3
MATH500	84.2	85.4
Average	**63.8**	**64.4**

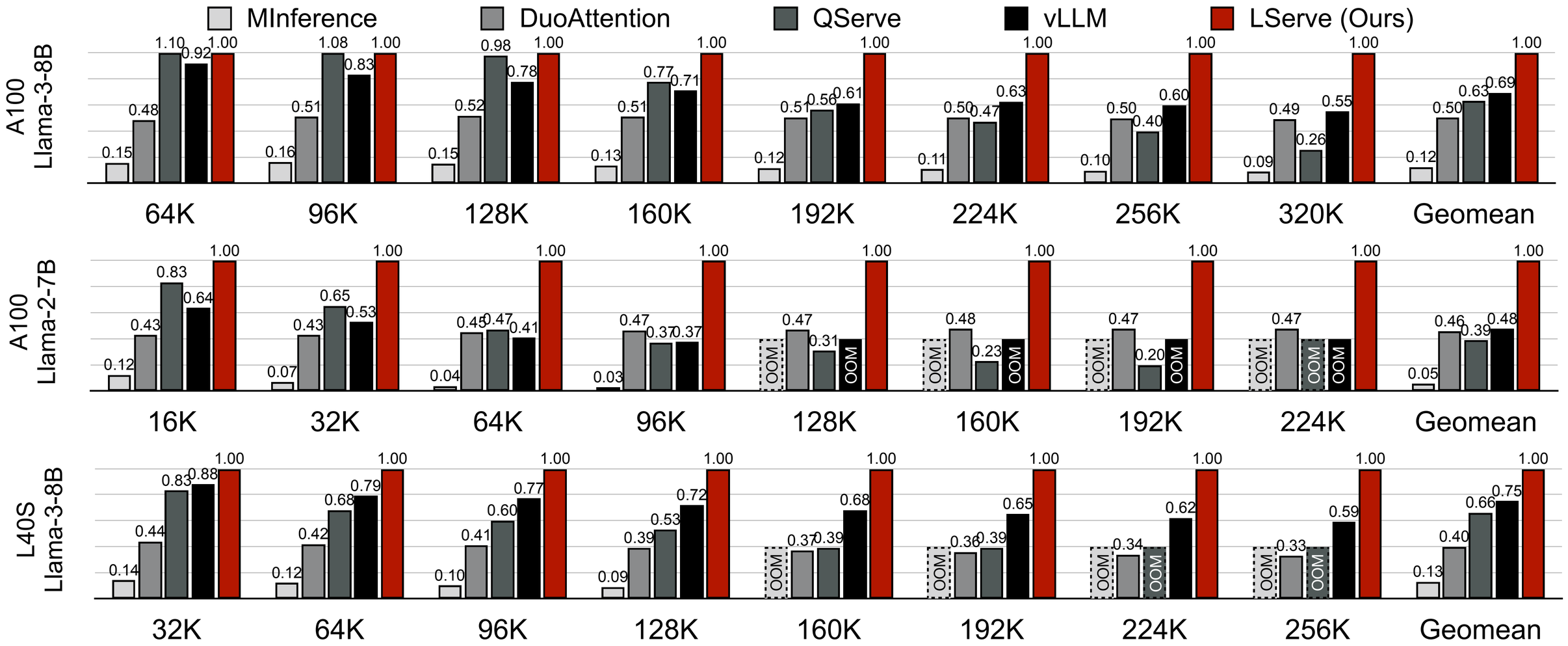
(c) Accuracy benchmark on reasoning tasks

Llama-3-8B	32K	64K	128K	160K	192K	256K
Dense	90.5	86.8	83.8	79.3	79.6	79.4
LServe	91.0	85.6	81.0	79.0	76.1	75.7

(b) LServe on RULER benchmark

Efficiency Benchmarks

LServe accelerates long-sequence decoding by 1.3-2.1x over vLLM

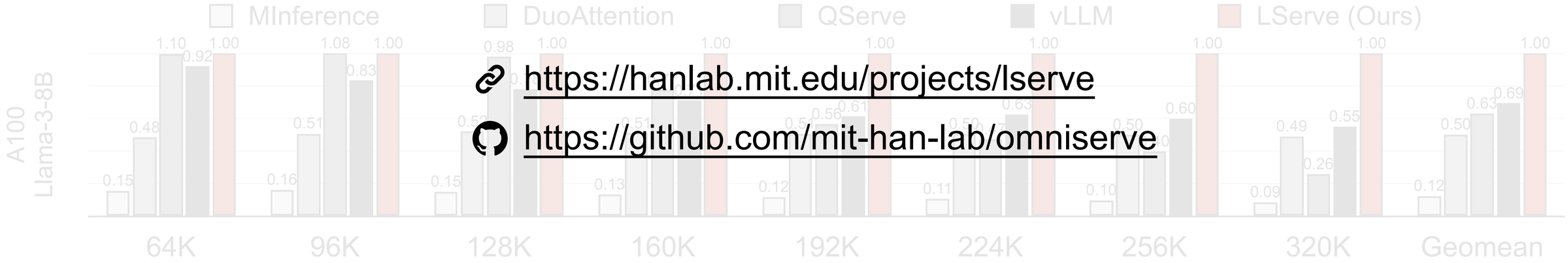


- Unified static and dynamic sparse attention
- Hardware-aware sparse paging system design
- LServe accelerates long-sequence decoding

Thank you!

- https://github.com/mit-han-lab/omniserve

LServe: Efficient Long-Sequence LLM Serving



https://hanlab.mit.edu/projects/lserve

https://github.com/mit-han-lab/omniserve