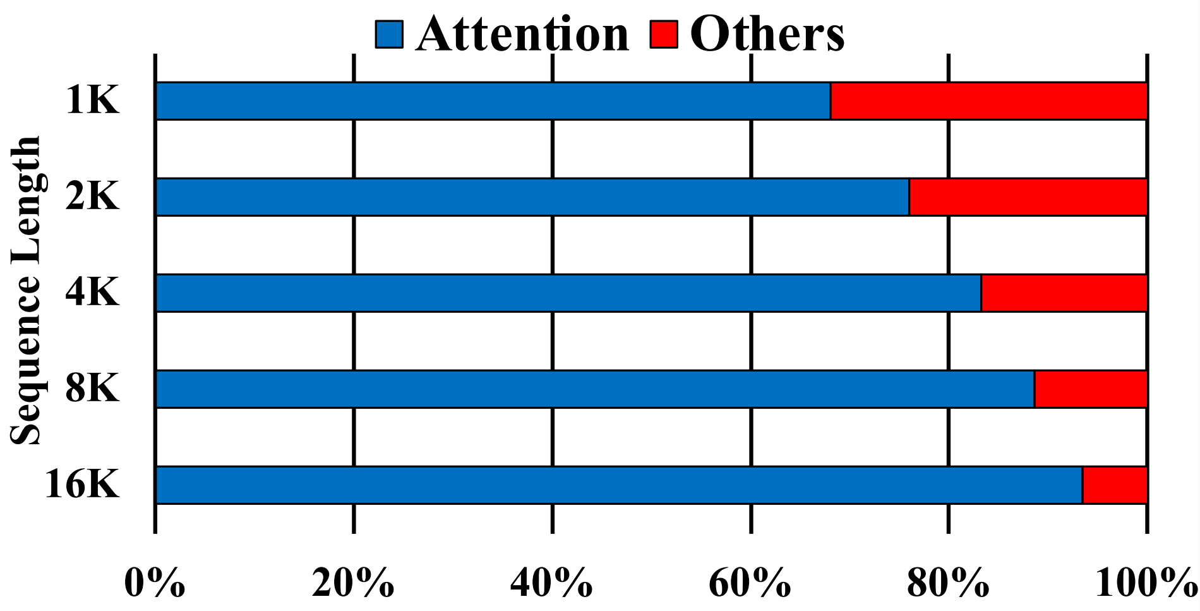
I/O Analysis is All You Need: An I/O Analysis for Long-Sequence Attention

Xiaoyang Lu[1], Boyu Long[2,3], Xiaoming Chen[2], Yinhe Han[2], Xian-He Sun[1]
[1]Illinois Institute of Technology, [2]Chinese Academy of Sciences, [3]University of Chinese Academy of Sciences

Long-Sequence Attention is the Bottleneck

Attention is the core operation in Transformer-based LLMs

- Inputs: Q, K, V
- O = Softmax (QK^T) V
- **Compute and memory cost scale quadratically with sequence length**



- Attention becomes the dominant runtime cost
- Naïve execution incurs repeated data movement (I/O) between on-chip memory and off-chip HBM.

Prior Studies

FlashAttention [NIPS'22]

- Uses block-wise online softmax to avoid materializing the full score matrix
- Reduces HBM traffic by tiling attention into on-chip blocks
- Still relies on **heuristic tiling,** not fully utilized the on-chip memory

FLAT [ASPLOS'23]

- Uses a fused row-granularity dataflow to keep more intermediates on chip
- Reduces I/O for softmax, but storing long rows on chip
- With long sequences, row storage can **increase I/O again**

I/O Analysis is Needed

I/O analysis provides principled answers to three key questions

- **I/O lower bound:** What is the minimum I/O under a given on-chip memory budget?
- **Optimal tiling:** What tile sizes minimize data movement and maximize on-chip memory utilization?
- **Practical scheduling:** What scheduling strategy realizes this lower bound in practice?

I/O Analysis: Foundation

Red-Blue Pebble Game
[Hong, Kung. 1981]

CDAG abstraction

- Vertex: data entry or intermediate result
- Edge: data dependency

Subcomputation

- One local region of CDAG

Dominator set (D_r)

- Minimum inputs needed

Red Pebble ●●●

- Data in fast memory

Blue Pebble ●●●●● ... ●

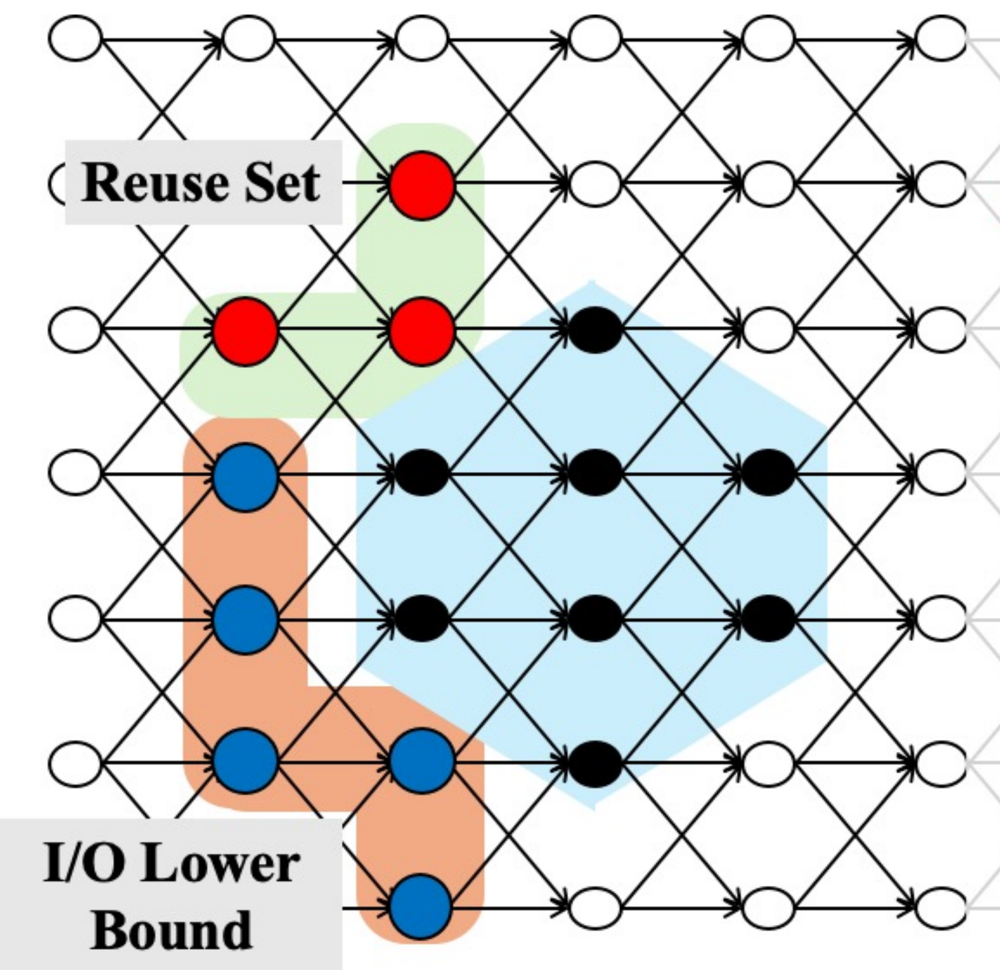
- Data in slow memory

Reuse Set (R_r)

- Data need not be loaded again if they can be reused

Store Set (W_r)

- Data must be stored back to slow memory.



I/O lower bound
Maximum reuse size

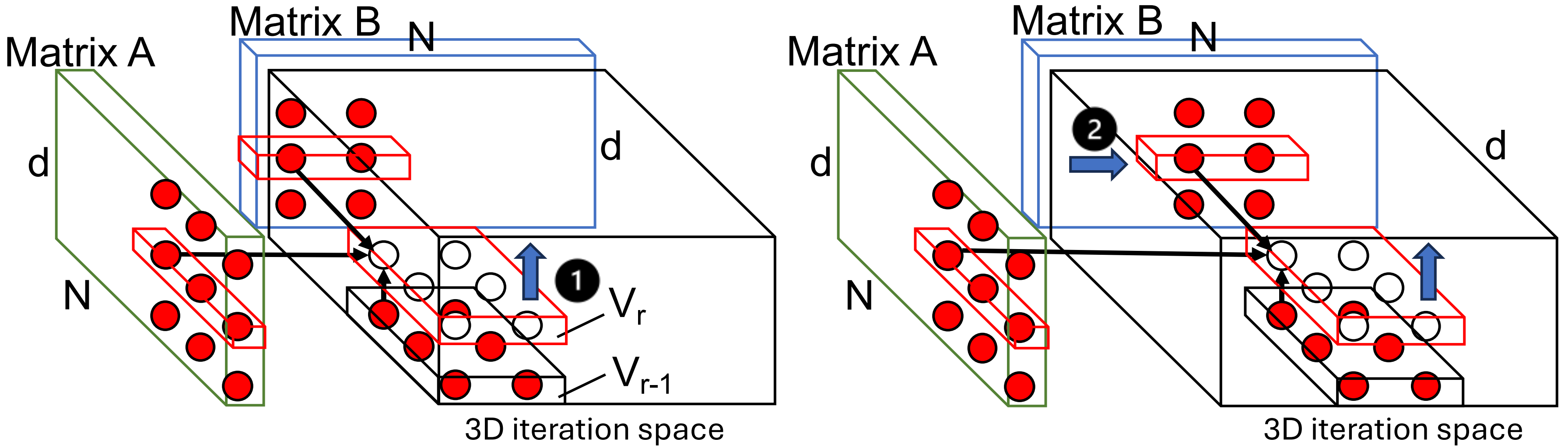
$$T_r \geq |D_r| - |R_r| + |W_r|$$

Dominator set size
Minimum store size

I/O Analysis: Tall-and-Skinny MMM

What we analyze

- **Tall-and-Skinny MMM:** C = AB, where N ≫ d, under on-chip memory budget M.
- **Immediate reuse:** keep partial outputs on chip for direct reuse by subsequent subcomputations, avoiding write-back.
- **Future reuse:** keep one input block on chip until it has been fully reused.
- **Objective:** maximize the compute-to-I/O ratio under realistic memory constraints.

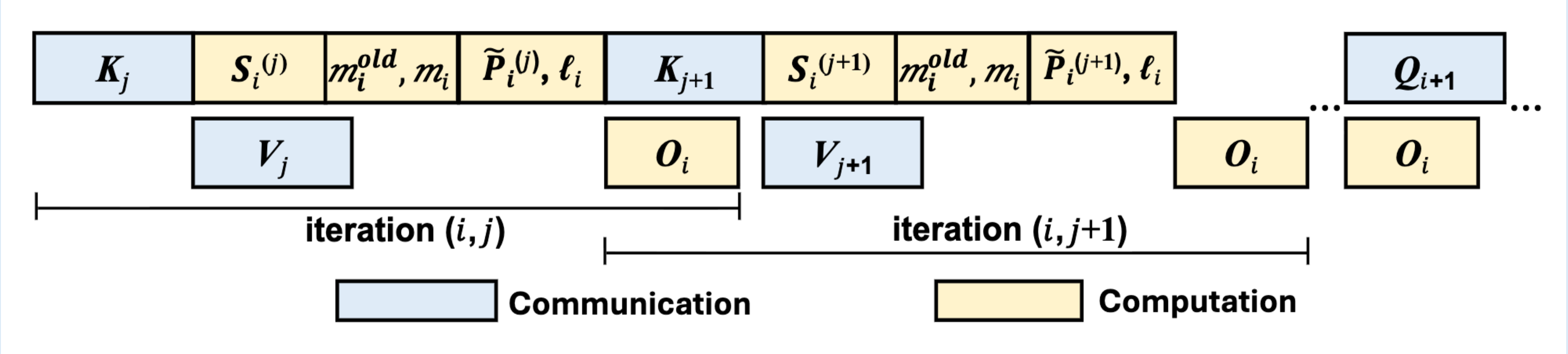


Maximize immediate reuse; minimize stores
Keep the A tile on chip for future reuse

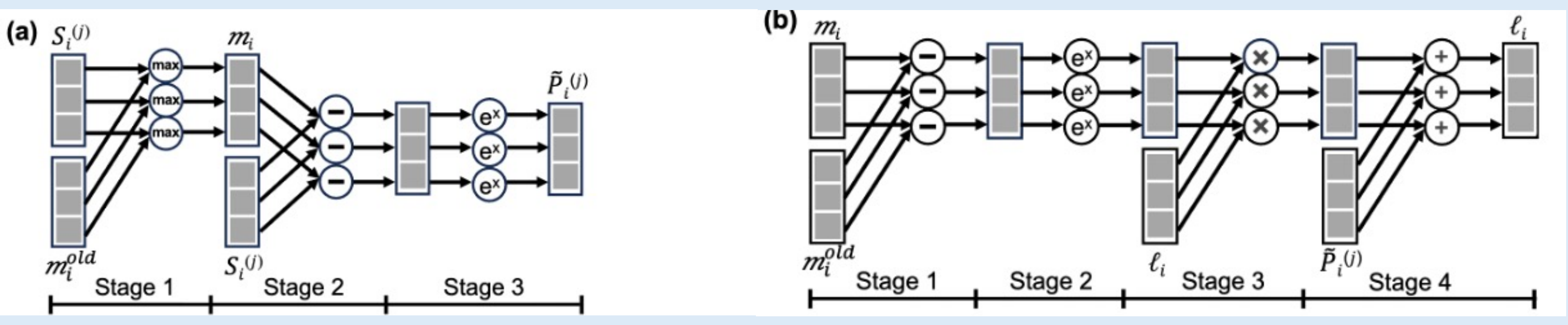
Derive tile sizes to maximize the compute-to-I/O ratio
Optimal tall-and-skinny MMM I/O scales as **$O(N^2 d^2/ M)$**

AttenIO Accelerator

- **I/O-Optimal Dataflow:** Derive tiling and scheduling for exact long-sequence attention
- **Three-Level Overlap:** Hide I/O stalls with fine-grained communication-computation overlapping

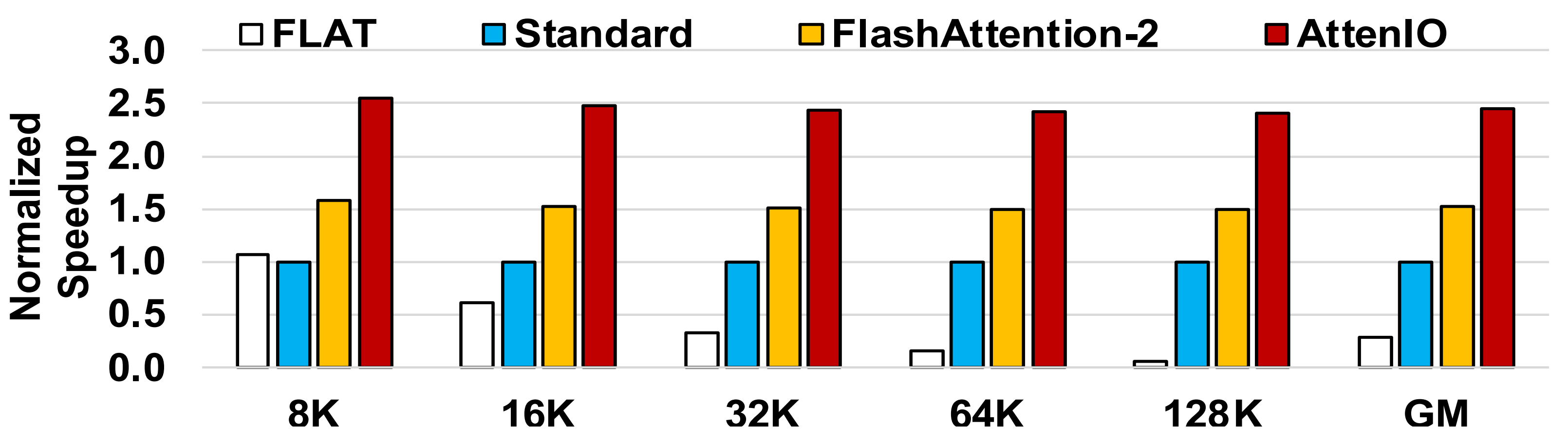


- **Parallel Softmax:** Exploit parallel patterns for efficient softmax execution

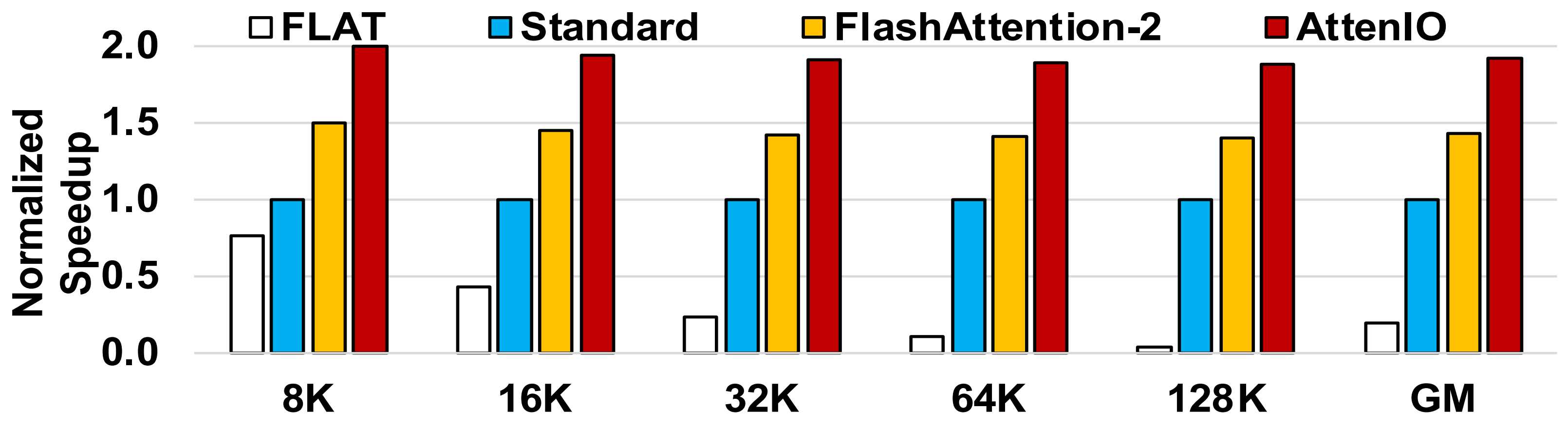


Results

Head dimension 64



Head dimension 128



Head dimension 64:
8.8× over FLAT
2.5× over Standard
1.6× over FlashAttention-2

Head dimension 128:
9.9× over FLAT
1.9× over Standard
1.3× over FlashAttention-2

AttenIO consistently outperforms all baselines across sequence lengths and head dimensions.