Breaking the Exactness Barrier

Interleaved DeepSeek Sparse Attention(IDSA)

Efficient inference for long-context reasoning

Yifan Guo

During Internship. Joint work with Wei Cui, Microsoft Research Asia

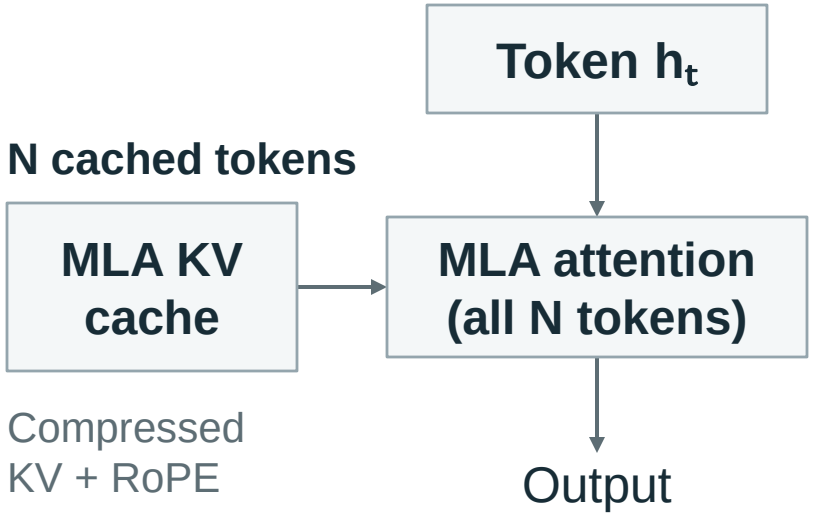
23 October 2026

Attention in DeepSeek-V3 and V3.2

MLA compresses KV representations. DSA adds token selection.

DeepSeek-V3

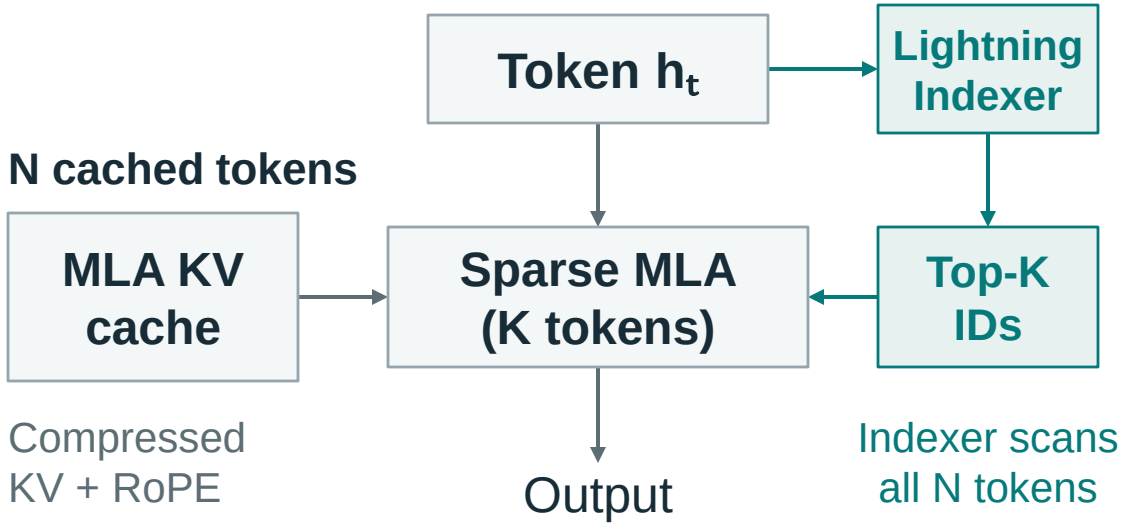
MLA over the full causal context



Smaller KV storage per token

DeepSeek-V3.2

DSA instantiated on MLA

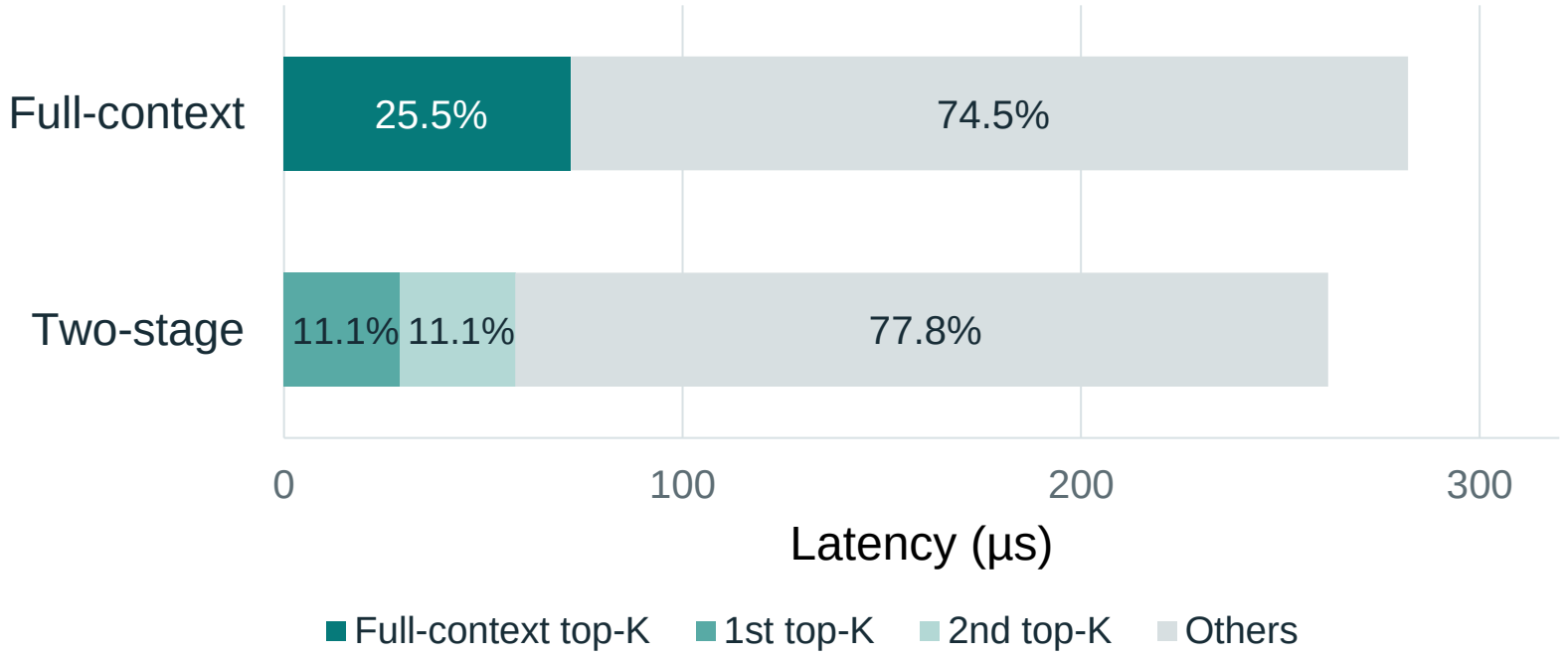


Fewer tokens in attention

Simplified decoding view. Projections and indexer key cache omitted. Sources: DeepSeek technical reports.

The distributed exactness barrier

Exact global top-K adds latency in both distributed strategies.



22.2%

of two-stage latency comes from top-K

11.1% local selection
11.1% global refinement

Full-context Each GPU repeats top-K over all N tokens.

Two-stage Local top-K, gather G·K candidates, then global top-K.

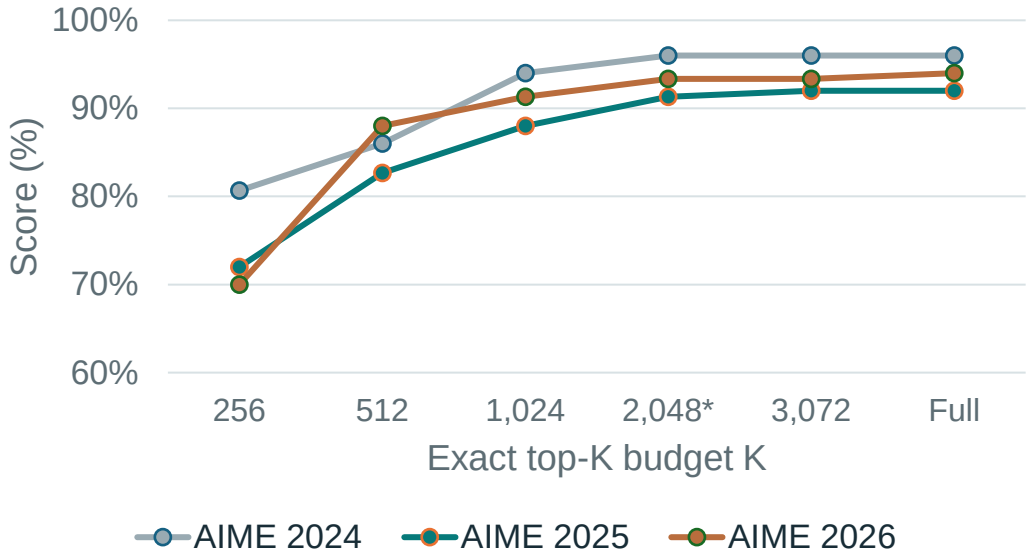
DeepSeek-V3.2, 8 × MI300X, 128K context, batch 1. Reconstructed from paper Fig. 1; latencies approximate.

The exact top-K budget can exceed 2,048

Vary the exact global top-K budget K with model weights unchanged.

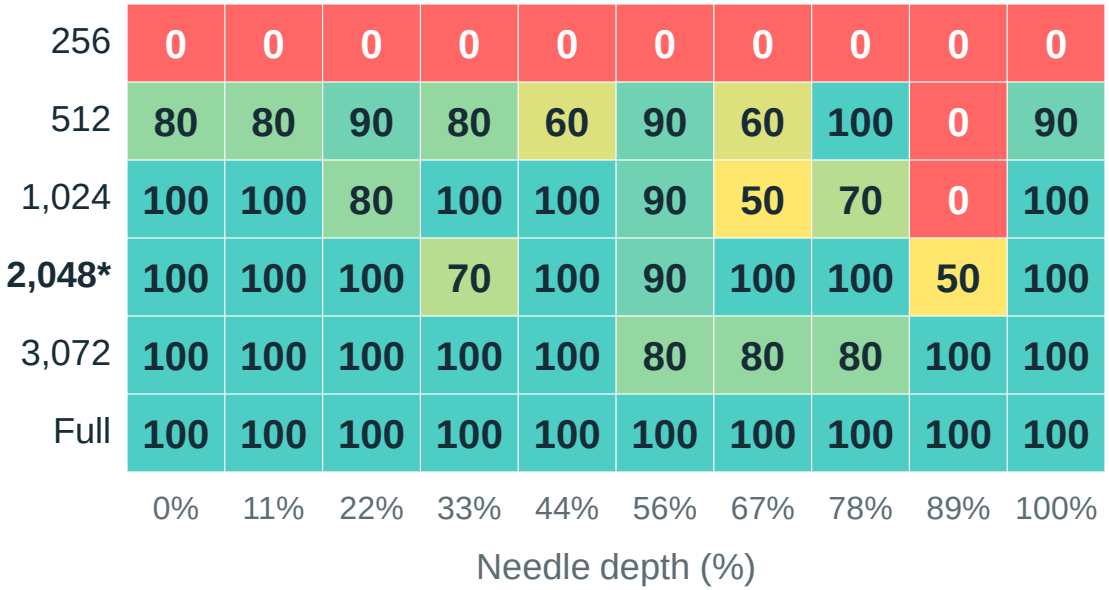
Mathematical reasoning

AIME 2024 / 2025 / 2026



Needle in a Haystack

128K context, accuracy (%)

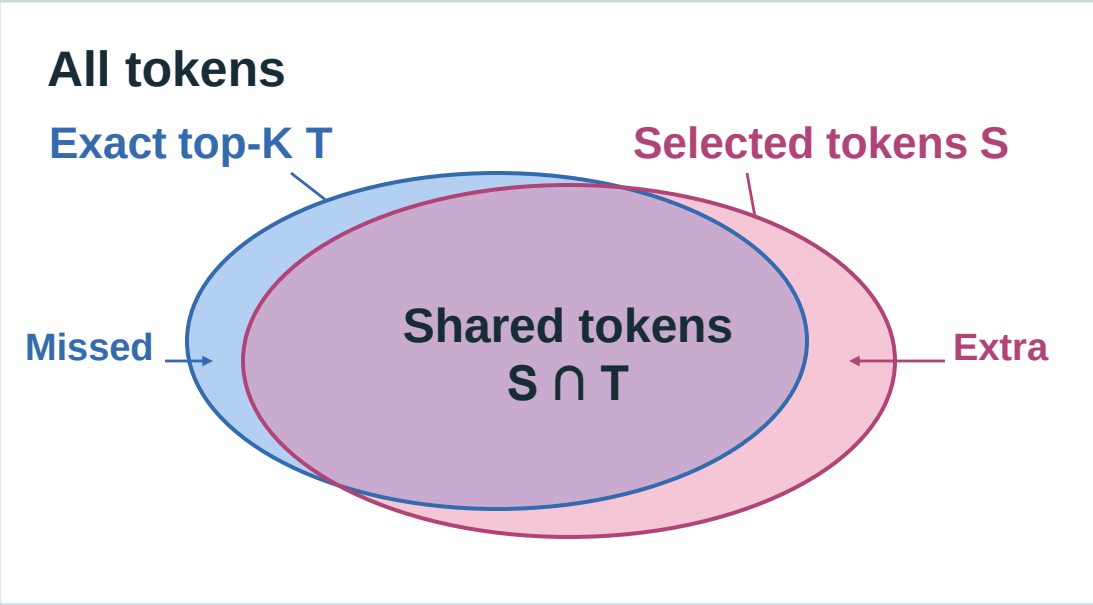


Extra context preserves reasoning scores and can improve retrieval.

AIME: 5 runs, thinking mode, no tools. Both tasks: temperature 1.0. * Default training budget. NIAH data: supplied screenshot.

IDSA uses independent local selection

m: tokens selected per GPU. K: exact global target. G: number of GPUs.



Shrink both differences

T \ S: missed top-K tokens

S \ T: extra selected tokens

Minimize local budget

$$\min_{m \in \mathbb{N}} \quad m$$

$$\text{s.t.} \quad \Pr[T \subseteq S(m)] \geq \gamma$$

$$m \geq \frac{K}{G}$$

$$T = \mathrm{Top}(C, K)$$

$$S(m) = \bigcup_{g=0}^{G-1} \mathrm{Top}(C_g, m)$$

γ: target probability of full top-K recall

Candidate gathering or shard-local attention can implement the union. Other collectives may remain.

Why interleaving balances candidate load

Three ways to assign the same 16 token positions to 4 GPUs.

Token positions	0	1	2	3	**4**	**5**	**6**	**7**	8	9	10	11	12	13	14	15

■ Important tokens (positions 4–7)

Contiguous

Fixed consecutive blocks

GPU	Token positions				Load
0	0	1	2	3	**0**
1	**4**	**5**	**6**	**7**	**4**
2	8	9	10	11	**0**
3	12	13	14	15	**0**

A cluster can concentrate on one GPU.

Random

Stochastic assignment

GPU	Token positions				Load
0	1	**4**	**7**	14	**2**
1	0	9	12	15	**0**
2	2	**5**	10	13	**1**
3	3	**6**	8	11	**1**

New seed, new assignment and candidate loads.

Interleaved

Deterministic: GPU = t mod 4

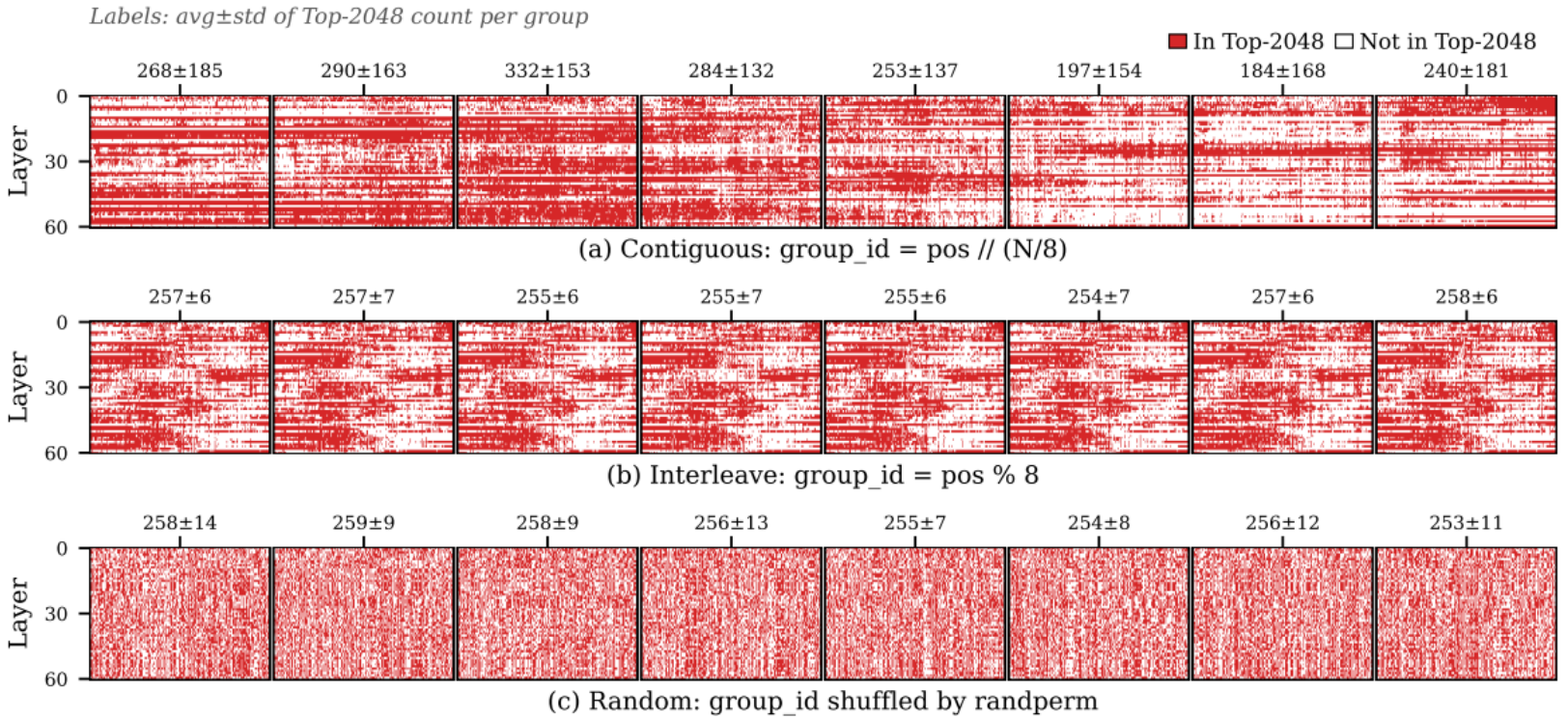
GPU	Token positions				Load
0	0	**4**	8	12	**1**
1	1	**5**	9	13	**1**
2	2	**6**	10	14	**1**
3	3	**7**	11	15	**1**

Fixed round-robin mapping spreads this cluster evenly.

Illustrative example. Load counts important tokens. Random shows one sampled permutation.

Interleaving lowers the observed imbalance

DeepSeek-V3.2, 61 layers and 10 prompts, K = 2,048, G = 8



128K context	Std.	Max load
Contiguous	318.3	1,766
Random	14.9	320
Interleaved	**11.7**	**311**

Heatmap: 4K example. Table: 128K statistics. Full-recall thresholds: Appendix A, Table 7.

From full recall to a binomial tail bound

Assume independent, uniform assignment of each token in the fixed global top-K set.

Load model

$$X_i = |T \cap C_i| \sim \text{Binomial}(K, 1/G)$$

Full recall

$$T \subseteq S(m) \iff \max_i \; X_i \le m$$

Union bound

$$\Pr[T \subseteq S(m)] = 1 - \Pr\Big(\bigcup_i \{X_i > m\}\Big)$$

$$\ge 1 - \sum_i \Pr(X_i > m) = 1 - G\,\Pr(X_1 > m)$$

Target probability

$\Pr[T \subseteq S(m)] \ge \gamma$ is ensured by

$$\Pr(X_1 > m) \le (1-\gamma)/G \iff F_X(m) \ge p := 1 - (1-\gamma)/G$$

Disjoint shards and consistent score ordering. The union bound does not require independent GPU loads.

From the tail bound to the local budget

Let $p = 1 - (1-\gamma)/G$ and $z_p = \Phi^{-1}(p)$, the standard normal p-quantile.

Exact threshold

$$m_{ex} = \min_{m} \{m \in \mathbb{N} : F_X(m) \geq p\}$$

$$F_X(m) = \sum_{j=0}^{m} \binom{K}{j} (1/G)^{j} (1-1/G)^{K-j}$$

Normal model

$$\mu = K/G, \quad \sigma = \sqrt{[K(G-1)]/G}$$

$$F_X(m) \approx \Phi((m + ½ - \mu)/\sigma) \geq p \quad \Rightarrow \; m \gtrsim \mu + z_p\sigma - ½$$

One-token margin

$$m_{ap} = \lceil \mu + z_p\sigma - ½ + 1 \rceil = \lceil \mu + z_p\sigma + ½ \rceil$$

32-token rounding

$$m = 32\lceil m_{ap}/32 \rceil$$

K = 2,048, G = 8, γ = 99%

$$m_{ex} = m_{ap} = 302, \quad m = 320 \text{ tokens/GPU}$$

m_{ex} is the minimum sufficient union-bound budget. Deterministic interleaving is evaluated empirically.

A closed-form budget rule for 8 GPUs

K = 2,048, G = 8, γ = 99%. Normal approximation with one token of margin.

$$m_{ap} = \left\lceil \frac{K}{G} + z_p \frac{\sqrt{[K(G-1)]}}{G} + ½ \right\rceil \qquad z_p = \Phi^{-1}(p), \quad p = 1 - (1-\gamma)/G$$

$$m_{ap} = \lceil 256 + 3.0233 \times 14.9666 + 0.5 \rceil = 302$$

Before alignment: m_{ap} = 302 tokens per GPU

Local budget m (tokens / GPU)

320

$m = 32\lceil 302/32 \rceil$

Equivalent counts for m = 320

ε = m − K/G = 64 extra tokens / GPU

Gm = 2,560 tokens in total (+25% over K)

Exact binomial inversion also gives 302. Random-assignment reference model; interleaving is evaluated empirically.

A small budget change removes global refinement

K = 2,048, G = 8. IDSA local budget: m = 320 tokens/GPU.

Per decoding query	Two-stage exact selection	IDSA (m = 320)
Selected per GPU	K = 2,048	m = 320
Candidates gathered (all GPUs)	G·K = 16,384	Gm = 2,560
Second global top-K	Required	Removed
Tokens used by attention	K = 2,048	Gm = 2,560

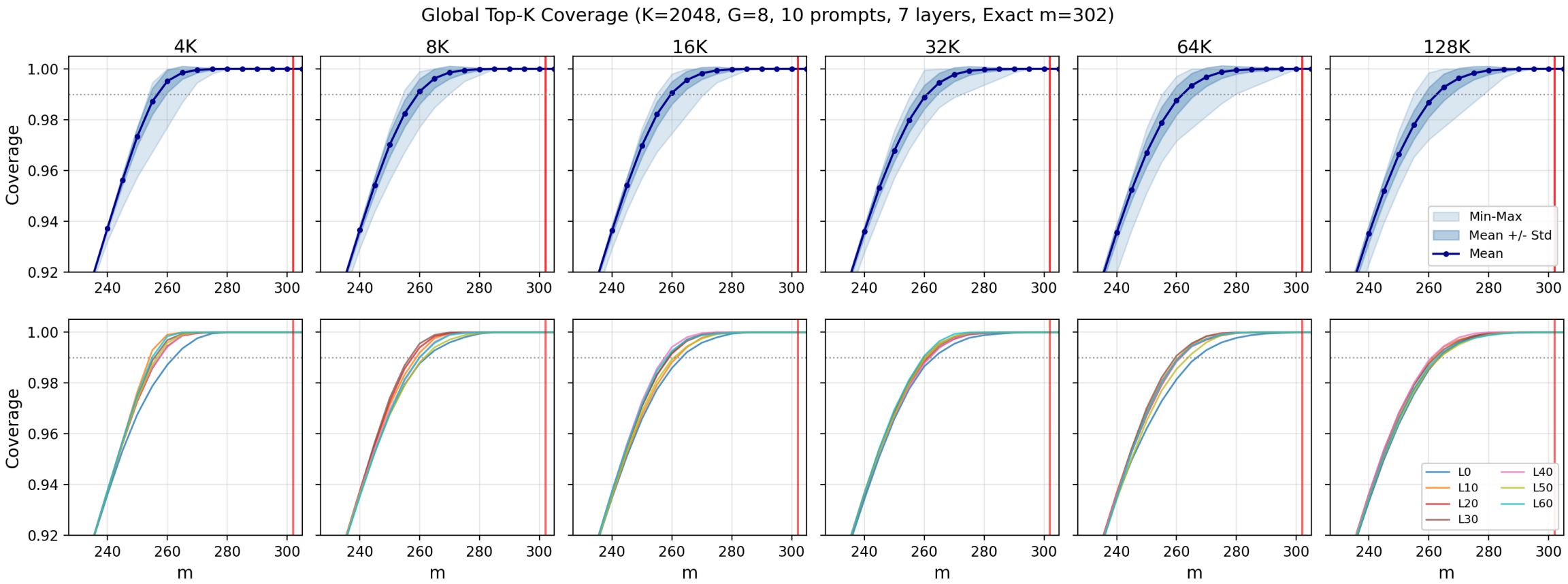
6.4× fewer candidate entries to gather

25% more attention candidates

Candidate counts compare against two-stage exact top-K. Byte traffic also depends on payload format and implementation.

Empirical coverage under interleaved partitioning

Global top-K token recall across 4K–128K contexts (K = 2,048, G = 8).



m: tokens/GPU. Red line: m = 302. Top: mean and spread. Bottom: layer curves.

Paper Fig. 5, Appendix A. Coverage is token recall, not the probability of full recall.

A moderate margin preserves AIME scores

IDSA at m = 320 tokens per GPU preserves AIME scores (G = 8).

DeepSeek-V3.2 (%)

Configuration	2024	2025	2026
Exact K = 2,048	96.0%	91.3%	93.3%
IDSA m = 256	96.0%	90.0%	92.0%
IDSA m = 320	**96.0%**	**91.3%**	**93.3%**
IDSA m = 512	96.0%	92.0%	94.0%

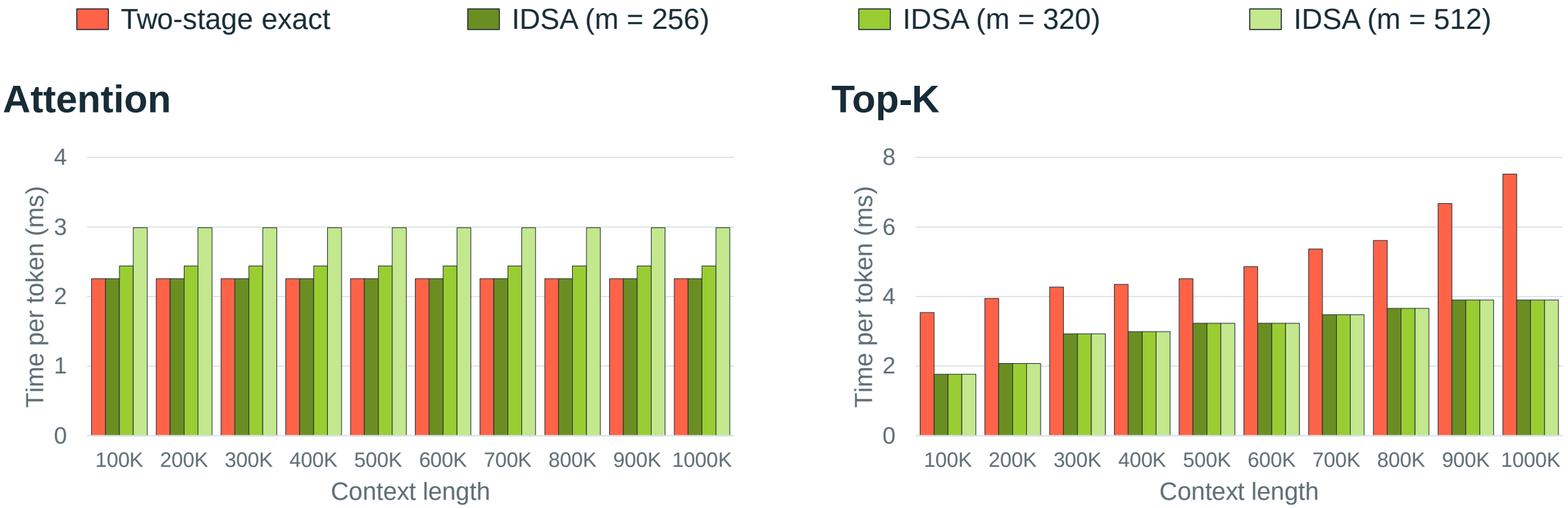
GLM-5 (%)

Configuration	2024	2025	2026
Exact K = 2,048	94.0%	93.3%	94.7%
IDSA m = 256	93.3%	92.7%	94.0%
IDSA m = 320	**94.0%**	**93.3%**	**95.3%**
IDSA m = 512	96.7%	96.0%	95.3%

Scores (%) = aggregate / 150 × 100. Five runs, temperature 1.0, thinking mode, no tools. No retraining.

Attention and Top-K latency

DeepSeek-V3.2, 8 × AMD MI300 GPUs (XGMI), K = 2,048. IDSA: m tokens/GPU.



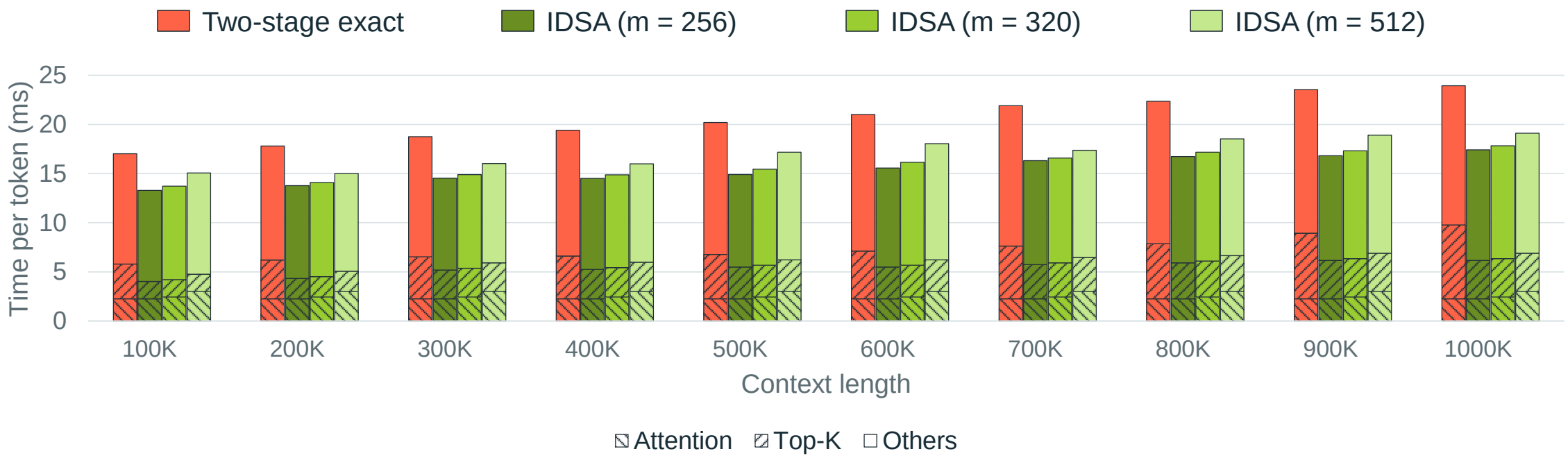
Attention cost stays nearly flat.

IDSA removes global refinement.

Latency reconstructed from the original vector PDF. m is the IDSA local selection budget, in tokens per GPU.

Decoding latency scales better with IDSA

DeepSeek-V3.2, 8 × AMD MI300 GPUs (XGMI), K = 2,048. IDSA: m tokens/GPU.



< 20 ms / token at the 1M-token systems test

Reconstructed from the source PDF. Million-token measurements test systems scaling, not million-token task quality.

Limitations

Conclude with one sentence.

