Breaking the Exactness Barrier: Interleaved DeepSeek Sparse Attention for Efficient Long Context Reasoning

Yifan Guo
Shanghai Jiao Tong University
yfguo26@cse.cuhk.edu.hk

Wei Cui
Microsoft Research Asia
weicu@microsoft.com

Abstract

Token-level dynamic sparse attention exemplified by DeepSeek Sparse Attention (DSA) selects the globally most relevant key-value tokens via an exact Top-K operator, achieving superior model quality over block-level alternatives. However, this exact selection creates a severe distributed inference bottleneck: enforcing an exact global Top-K across GPUs inevitably incurs either redundant full-context retrieval or costly multi-stage cross-device synchronization, which largely negates the computational advantages of DSA at long context lengths. We first show that the exact Top-K bound is unnecessary during inference: once the truly critical tokens are recalled, admitting additional context can preserve or potentially improve accuracy. Leveraging this insight, we propose *Interleaved DeepSeek Sparse Attention* (IDSA), which distributes tokens across GPUs in an interleaved layout so that each device performs only a relaxed local Top-m selection. Under this layout, the union of independent per-GPU Top-m selections near-completely covers the globally most relevant Top-K tokens. This allows each device to proceed with its local selection with minimal cross-GPU overhead while avoiding both expensive full-context Top-K computation and multi-stage cross-GPU merging, enabling a not only distributed but also synchronization-efficient inference pipeline. Without any retraining, IDSA delivers substantial efficiency gains in sparse token selection, particularly for long sequences exceeding 100K tokens while preserving generally comparable task performance on DeepSeek-V3.2 and GLM-5 series across AIME, NIAH, and selected RULER tasks.

1 Introduction

Long chain-of-thought reasoning[1–3], multi-turn dialog scenarios[4, 5] and agentic AI[6–8] are driving Large Language Models (LLMs) toward ever-longer contexts[9–11], where the computational cost of full attention has become a critical bottleneck. To address this, efficient attention mechanisms have primarily evolved along two branches: linear attention[12–14] and sparse attention. Sparse attention can be further categorized into fixed patterns, such as Sliding Window Attention (SWA)[15–17], and dynamic retrieval attention. Because linear attention and SWA inherently possess limited memory capacity and permanently forget distant historical information, they struggle with long-context complex reasoning tasks. In contrast, dynamic retrieval sparse attention preserves whole context information and adaptively select critical information. Within this paradigm, Token-level fine-grained DeepSeek Sparse Attention (DSA) [18] exhibits exceptional performance, leading to its successful application in massive models like DeepSeek-V3.2 ($\sim$700B parameters).

While previous dynamic retrieval schemes[19–26] often operate at a coarse block-level granularity to reduce overhead, DSA's token-level fine-grained approach yields superior model quality. However, this fine granularity drastically increases algorithmic complexity and suffers from low hardware utilization. Specifically, the presence of the Top-K operation in the DSA Lightning Indexer causes the actual inference throughput during reasoning to frequently drop below that of full attention, despite

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

the theoretical computation being far lower (as will be shown in Figure 4). Since the generative decoding phase can dominate the overall latency and computational cost in long-context applications, we focus our optimization and efficiency analysis primarily on decoding.

In this paper, we introduce a controlled over-selection strategy for Top-K that targets high recall of the reference Top-K set while significantly reducing the computational overhead and parallelization barriers of exact global Top-K selection. Our contributions are summarized as follows:

- We observe that an exact Top-K selection is not strictly necessary during inference in Section 2.2. Although the DSA architecture is continuously pretrained with a rigid $K = 2048$ selection, our experiments show that increasing K beyond this default maintains or even improves model performance.
- We propose Interleaved DeepSeek Sparse Attention (IDSA). An interleaved layout distributes tokens uniformly across GPUs. Each device performs a local Top-m selection ($m = K/G+\epsilon$ where ϵ is a small margin), and the randomized-layout analysis provides a sufficient budget for full Top-K recall with at least 99% probability (Section 3.4), while the deployed interleaved layout is evaluated empirically (Section 3.5).
- We analyze the throughput achieved by different Lightning Indexer configurations in Section 4.3. We demonstrate that without any parameter updates, IDSA achieves substantial decoding throughput improvement, demonstrating effective algorithm-system co-design.

We provide a docker-distributed inference service with the IDSA implementation source included in the image. Details of code availability, service setup, and the evaluation interface are documented in Appendix F.

2 The Cost of Exactness

2.1 The Bottleneck of Top-K for DSA

DeepSeek Sparse Attention (DSA) employs a lightweight *Lightning Indexer* module that predicts approximate attention scores (i.e., context-importance scores), and performs an explicit Top-K selection over key-value tokens for each query. While other dynamic retrieval schemes often operate at a coarse block-level granularity (e.g., size 64), DSA operates Top-K selection on the full token sequence. This fine granularity yields superior model quality but significantly amplifies the computational cost: exact token-level Top-K selection incurs substantial latency despite the reduced attention workload (Figure 1).

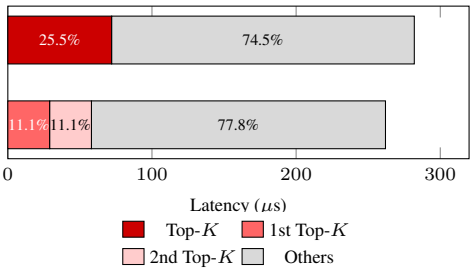

Figure 1: Latency breakdown of exact Top-K multi-GPU implementations. **Top bar (Full-Context):** Each GPU independently computes Top-K over the full global context, leading to prohibitively high retrieval costs. **Bottom bar (Two-Stage):** A two-stage strategy reduces individual Top-K overhead but introduces a secondary merge phase. **Setup:** Experiments are conducted on DeepSeek-V3.2 using $8\times$ MI300X GPUs at 128K context length with batch size $= 1$.

Computing TopK in multi-GPU environments presents a fundamental challenge: it is difficult to simultaneously avoid both the expensive cross-GPU synchronization overhead and the local TopK retrieval cost. If we aim to minimize GPU synchronization overhead, each GPU must independently compute TopK over the entire global context, which results in prohibitively high computational costs (See Figure 1 Top). Conversely, to reduce the overhead of each GPU computing TopK over the full context, we must adopt at least a two-stage TopK merging strategy: first, each GPU independently computes TopK over a local partition of the context, results are then communicated and merged through a second TopK operation to obtain the final result. However, the combined overhead of this approach remains substantial (see Figure 1 Bottom), making it difficult to achieve noticeable improvement.

Benchmark	Top-256	Top-512	Top-1024	Top-2048*	Top-3072	Full
AIME-24	121	129	141	144	144	144
AIME-25	108	124	132	137	138	138
AIME-26	105	132	137	140	140	141

Table 1: Model performance of DeepSeek-V3.2 (thinking mode, no tools) under different inference-time Top-K configurations on the AIME benchmarks. Scores are aggregated with temperature = 1.0 over 5 independent runs. * indicates the default configuration during contiguous pretraining.

2.2 Insights from Exact Top-K to Over-token Selection

During the continuous pretraining of DSA-architecture models, the backbone parameters and the Lightning Indexer are jointly optimized under a fixed configuration where exactly $K = 2048$ tokens are selected. Intuitively, to achieve optimal performance during inference, K should be strictly constrained to 2048. However, our practical deployments and experiments suggest that this constraint can be relaxed.

We evaluated performance on long reasoning mathematical tasks such as AIME-24, AIME-25 and AIME-26, as well as retrieval tasks like Needle In A Haystack (NIAH). As shown in Table 1 and Figure 2, using the unmodified pretrained DeepSeek-V3.2 weights, we sequentially varied the Top-K value at inference time across 256, 512, 1024, 2048, 3072, and the full sequence length (i.e., full attention), where 2048 corresponds to the training-time configuration. We find that moderate over-selection in Top-K ($K > 2048$) yields comparable or superior performance, provided that the critical token from Top-2048 remains included. Based on this observation, IDSA is motivated by the need for a parallel-friendly selection of top tokens with perfect recall of the Top-K set while minimizing redundant selections. This mechanism can greatly improve the parallel efficiency of global token selection while leveraging over-token selection to ensure that the originally-designed Top-2048 token is fully preserved. We hypothesize that this tolerance or even preference for denser context stems from the standard training recipe of DSA-based models: they typically undergo initial pretraining with full attention before transitioning to DSA for continuous pretraining. Consequently, the backbone inherently retains the capacity to process richer contexts beyond the strict algorithmic constraint of K. The experimental results in Section 4.1 and Section 4.2 further demonstrate the existence of this phenomenon.

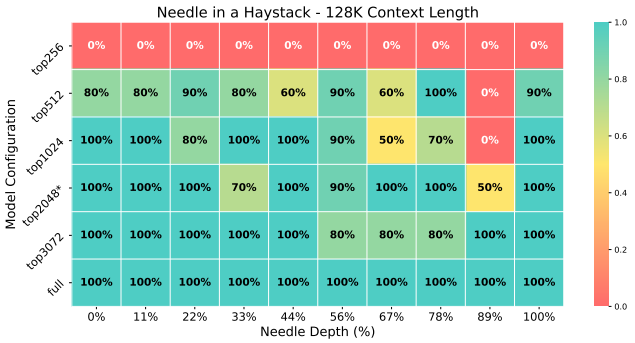

Figure 2: DeepSeek-V3.2 on the 128K Context Needle-in-a-Haystack task under different Top-K settings with temperature = 1. * indicates the configuration used in contiguous pretraining.

3 Interleaved DSA

3.1 Mathematical Modeling of the Problem

The core idea of IDSA (Interleaved DSA) is to smartly partition the global context problem P into G completely independent sub-problems $P_0, \ldots, P_{G-1}$. Each sub-problem independently computes Top-m, and these results no longer require a second-stage Top-K refinement. In this way, IDSA effectively scatters the Top-K overhead across different GPUs, while it also prevents multi-round Top-K refinement and cross-GPU synchronization, subject to the condition:

$$\text{Top}(\mathcal{P}, K) \subseteq \bigcup_{g=0}^{G-1} \text{Top}(\mathcal{P}_g, m) \tag{1}$$

where the ideal lower bound of m is K/G, and the larger the value of m, the more likely the expression holds. Hence, how to minimize m becomes the important next step.

3.2 Top-K Distribution in Partitioned Context

Theoretically, the key to minimizing m lies in achieving a balanced distribution of top-K elements across partitions, ensuring that no single partition is overloaded. Assuming a sequence of length N is partitioned into G groups (GPUs), each corresponding to an independent sub-problem, we explored 3 partitioning strategies for the context-importance scores computed by the standard DSA Lightning Indexer: *Contiguous*, *Interleaving*, and *Random*.

The **Contiguous** strategy assigns every consecutive N/G scores into the same group:

$$\mathcal{P}(\text{Contiguous})_g = \{\text{score}_t \mid \lfloor t/\lceil \frac{N}{G} \rceil \rfloor = g\}, \quad 0 \le g < G$$

The **Interleaving** strategy distributes scores using a round-robin approach, alternating among the G groups:

$$\mathcal{P}(\text{Interleaved})_g = \{\text{score}_t \mid t \bmod G = g\}, \quad 0 \le g < G$$

The **Random** strategy randomly assigns scores uniformly across the G groups:

$$\mathcal{P}(\text{Random})_g = \{\text{score}_t \mid r_t \bmod G = g\}, \qquad 0 \le g < G,\ (r_0, .., r_{N-1}) = \text{randperm}(N)$$

We evaluated these strategies on real data using 8 GPUs (Figure 3) visually, measuring how evenly the top 2048 globally important tokens are distributed across groups: Figure 3(a) visualizes context importance on DeepSeek-V3.2 at a sequence length of 4K. We found that context-importance scores exhibit strong continuity and clustering. Consequently, the Contiguous grouping strategy causes severe load imbalance across the GPUs, with massive dynamic variance between different layers. Figure 3(b) shows that the Interleaved layout effectively decomposes large clustered regions and maps them evenly across GPUs. Crucially, it preserves the **intrinsic temporal patterns** within each group as a **condensed** version of Figure 3(a). Conversely, the Random layout (Figure 3(c)) scatters tokens stochastically, achieving expected uniformity at the cost of damaged local structural coherence.

3.3 Variance Reduction via Continuity

Table 2 further compares these three layouts quantitatively. While the Contiguous layout suffers from massive load variances (Std) and extreme worst-case over-selection bounds (ϵ'), both Random and Interleaved layouts effectively mitigate this imbalance. Notably, the Interleaved strategy consistently outperforms the Random layout: it achieves the tightest absolute load bounds (lowest Max/ϵ') and minimal standard deviation across all context lengths. This empirical advantage is consistent with variance reduction via local continuity. Unlike the Random layout, which treats token assignment as independent stochastic events and inherently introduces **statistical variance and collisions**, the Interleaved strategy effectively functions as systematic sampling, where tokens are assigned to GPUs in a fixed round-robin pattern rather than randomly. Because real attention scores exhibit strong local continuity, contiguous blocks of highly-scored tokens are deterministically distributed across all G GPUs, yielding near-zero variance when block sizes align with G. For Random, each shard's share of the selected tokens approaches $1/G$ as the global context size grows, with the variance of this share decreasing proportionally to $1/K$.

3.4 Modeling the Upper Bound of Selection

Setup and random-assignment assumption. As established in Section 3.2, modeling the exact distribution of the Interleaved layout is challenging due to dynamic sequence variations. However, because we have demonstrated in Section 3.3 that the Interleaved strategy reduces load variance compared to independent stochastic assignment, we can model the Random layout as an analytical baseline for selecting the local budget of the Interleaved layout. While independent random

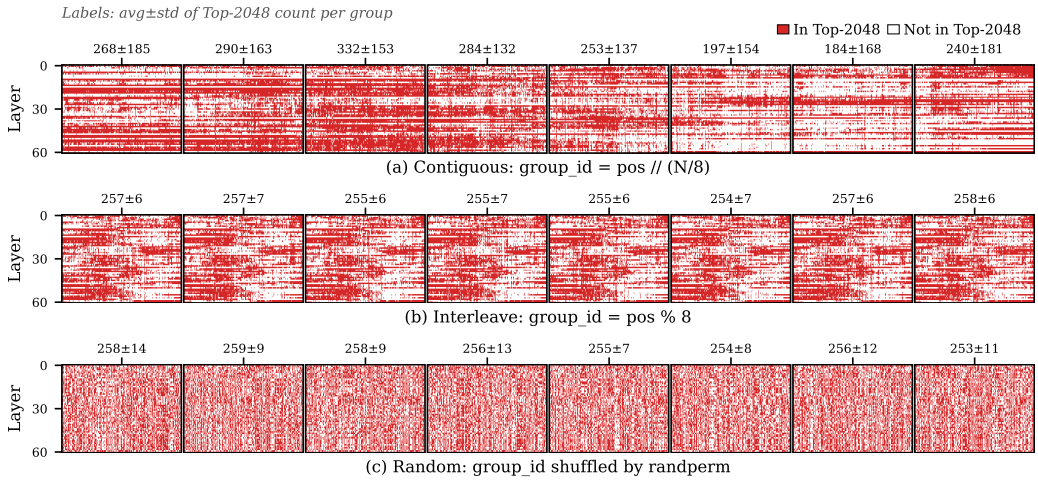

Figure 3: Distribution of DeepSeek-V3.2 context importance under different grouping strategies.

Seq Len	Min Load			Max Load			Std			ϵ'		
	Con.	Int.	Ran.	Con.	Int.	Ran.	Con.	Int.	Ran.	Con.	Int.	Ran.
4K	0	230	214	500	285	304	158.1	**6.1**	12.8	244	**29**	48
8K	1	223	205	997	295	304	187.0	**8.9**	13.6	741	**39**	48
16K	0	213	210	1119	293	317	203.6	**9.8**	14.1	863	**37**	61
32K	0	219	208	1910	297	307	291.8	**10.3**	13.4	1654	**41**	51
64K	0	204	202	1978	300	310	310.0	**11.1**	14.6	1722	**44**	54
128K	0	211	204	1766	311	320	318.3	**11.7**	14.9	1510	**55**	64

Table 2: Absolute per-partition Top-K load statistics under different layouts across 61 layers and 10 prompts ($K = 2048$, $G = 8$). **Min/Max** are the extreme numbers of tokens assigned to any single GPU. **Std** represents the spatial load variance (lower is better). ϵ' denotes the additional local selection budget required for full recall on all evaluated instances (Max $- K/G$).

assignment is not a strict mathematical worst-case model for deterministic interleaving, it provides a reference model whose sufficient selection budget is **empirically conservative** in our interleaved-layout experiments. Specifically, we assume that each token in the fixed global Top-K set is assigned independently and uniformly to one of G GPUs. Under this assumption, the number of Top-K tokens assigned to GPU i follows a binomial distribution:

$$X_i \sim \mathrm{Binomial}\big(K, \tfrac{1}{G}\big)\,. \tag{2}$$

We aim to select the local Top-m tokens on each GPU such that gathering them guarantees high recall of the exact global Top-K.

Guarantee via union bound. To ensure strict exactness, we want the gathered candidate set to contain *all* global Top-K tokens with probability at least $\gamma \in (0, 1)$:

$$\Pr\left(\max_{1\le i\le G} X_i \le m\right) \ge \gamma. \tag{3}$$

Using Boole's inequality (union bound) [27],

$$\Pr(\max_i X_i \le m) = 1 - \Pr\left(\bigcup_{i=1}^{G}\{X_i > m\}\right) \ge 1 - \sum_{i=1}^{G}\Pr(X_i > m) = 1 - G\Pr(X_1 > m). \tag{4}$$

Therefore it suffices to enforce $\Pr(X_1 \le m) \ge 1 - \frac{1-\gamma}{G}$, yielding the following theorem.

Theorem 3.1 (Distributed Top-K recall upper bound). *Let* $X \sim \mathrm{Binomial}(K, \frac{1}{G})$. *If*

$$\sum_{k=0}^{m_{Exact}} \binom{K}{k} \left(\tfrac{1}{G}\right)^k \left(1-\tfrac{1}{G}\right)^{K-k} \geq 1 - \tfrac{1-\gamma}{G}, \tag{5}$$

then gathering local Top-m *from each of the* G *GPUs contains the exact global Top-*K *tokens with probability at least* γ *under independent uniform distribution.*

Closed-form engineering rule (CLT + continuity correction). Direct evaluation of (5) can be numerically awkward for large K. Using a normal approximation with continuity correction [28, 29], for $X \sim \mathrm{Binomial}(K, \frac{1}{G})$,

$$\mu = \tfrac{K}{G}, \qquad \sigma = \tfrac{\sqrt{K(G-1)}}{G}. \tag{6}$$

Let Z_p denote the p-quantile of $\mathcal{N}(0,1)$. Adding one token of margin to the continuity-corrected normal approximation gives

$$m_{\text{Approx}} = \left\lceil \tfrac{K}{G} + Z_{1-\frac{1-\gamma}{G}} \cdot \tfrac{\sqrt{K(G-1)}}{G} + 0.5 \right\rceil. \tag{7}$$

This decomposes into an average load $\frac{K}{G}$ plus an uncertainty margin $Z\sigma$ capturing the redundancy needed to absorb random imbalance. Consequently, the total gathered candidates scale as

$$Gm_{\text{Approx}} \approx K + Z_{1-\frac{1-\gamma}{G}} \sqrt{K(G-1)}, \tag{8}$$

implying a sublinear (in G) redundancy growth and the total over-selection across the G GPUs.

Hardware-aligned rounding. For practical deployment, we round m to warp-friendly multiples:

$$m_{\text{eng}} = \left\lceil \tfrac{m_{\text{Approx}}}{32} \right\rceil \cdot 32. \tag{9}$$

To illustrate these derivations, Table 3 presents the calculated bounds for $K = 2048$, $G = 8$, and $\gamma = 0.99$. The theoretical safe threshold is calculated as 302.

Confidence (γ)	K/G	$\boldsymbol{m}_{\text{Exact}}$	$\boldsymbol{m}_{\text{Approx}}$	$\boldsymbol{m}_{\text{eng}}$
99.0%	256	302	302	320

Table 3: Theoretical local Top-m upper bounds calculated based on the independent binomial model.

Layout	$\boldsymbol{G=8}$		$\boldsymbol{G=4}$		$\boldsymbol{G=2}$		$\boldsymbol{G=1}$	
	m	$G\cdot\epsilon$	m	$G\cdot\epsilon$	m	$G\cdot\epsilon$	m	$G\cdot\epsilon$
Theoretical	302	368	568	224	1082	116	2048	0
Random	305	392	565	212	1079	110	2048	0
Interleaved	**290**	**272**	**553**	**164**	**1065**	**82**	2048	0

Table 4: Theoretical and empirical selection threshold m and over-selection ϵ under different numbers of GPUs G. The theoretical results are derived from Equation (5).

3.5 Empirical Validation

We empirically validate the required local selection threshold m for 99% lossless recall across various sequence lengths. Detailed quantitative results and full coverage trajectories are provided in Appendix Table 7 and Figure 5. The empirical thresholds for the Random layout tightly fluctuate around our calculated theoretical bound of $m = 302$, robustly confirming our Binomial modeling. Crucially, the Interleaved layout requires even lower thresholds (277∼294), validating its variance-reduction property. Our deployed hardware-aligned boundary of $m = 320$ provides additional margin above the measured recall thresholds.

DeepSeek-V3.2 Configuration	AIME 24	AIME 25	AIME 26
Standard-Top2048*	144	137	140
IDSA-Top2048 ($\epsilon = 0$)	144	135	138
IDSA-Top2560 ($\epsilon = 64$)	144	137	140
IDSA-Top4096 ($\epsilon = 256$)	144	138	141

GLM-5 Configuration	AIME 24	AIME 25	AIME 26
Standard-Top256	135	130	136
Standard-Top512	135	134	137
Standard-Top1024	138	135	140
Standard-Top2048*	141	140	142
IDSA-Top2048 ($\epsilon = 0$)	140	139	141
IDSA-Top2560 ($\epsilon = 64$)	141	140	143
IDSA-Top4096 ($\epsilon = 256$)	145	144	143
Full-Attention	145	144	144

Table 5: Performance of DeepSeek-V3.2 and GLM-5 on AIME benchmarks under different Top-K configurations. Scores are aggregated over 5 independent runs with temperature = 1.0 (thinking mode, no tools). * indicates default configurations.

3.6 GPU Scaling Efficiency

Scalability hinges not only on sequence length but also on the parallelism degree (G). In real-world serving, where NVLink/XGMI typically provides P2P connectivity within 8-device nodes, tensor parallelism on attention rarely exceeds the boundary of $G = 8$ as inter-node communication introduces latency that significantly degrades inference throughput. Instead, practitioners favor hybrid TP-DP for other scales. Following this convention, we analyze the scaling behavior when Top-K is partitioned within the practical bounds of $G \in \{1, 2, 4, 8\}$. As detailed in Table 4, both theoretical predictions and empirical results confirm that the total system-wide over-selection overhead ($G \times \epsilon$) strictly decreases as G shrinks. The Interleaved layout consistently maintains the lowest redundancy across all configurations—dropping its total system-wide extra tokens from 272 at $G = 8$ to merely 82 at $G = 2$. This confirms that IDSA is not only robust at the maximum intra-node parallel limit but becomes remarkably lightweight and resource-efficient when deployed on smaller GPU clusters.

4 Experiments

Under a train-free premise, IDSA not only preserves excellent model performance but also significantly improves model throughput, particularly at reasoning lengths exceeding 100K.

4.1 Performance on DeepSeek-V3.2

We evaluate DeepSeek-V3.2 under "thinking mode" and "no tools" at $temperature = 1.0$, with results aggregated over 5 independent runs. As shown in Table 5, the performance across AIME-24, AIME-25, and AIME-26 demonstrates robust stability across varied threshold settings.

IDSA-Top2048 matches the standard exact Top-2048 baseline on AIME-24 and shows only a marginal drop on the harder AIME-25/26 cases. Introducing a small over-selection margin ($\epsilon = 64$) fully recovers the baseline performance, while larger budgets ($\epsilon = 256$) consistently yield slight improvements, approaching the full-attention upper bound. These results indicate that relaxing strict Top-K exactness generally preserves reasoning performance and can even be beneficial.

4.2 Performance on GLM-5

We further examine whether the observed tolerance to over-selection extends to another DSA-based model. To ensure this finding generalizes beyond DeepSeek-V3.2, we further validate IDSA on the GLM-5 model ($\sim$750B parameters) [30]. As detailed in Table 5, GLM-5 demonstrates generally stable or improved scores as the selection budget increases across AIME-24, AIME-25, and AIME-26. Similar to DeepSeek-V3.2, IDSA completely matches or slightly surpasses the standard exact Top-2048 baseline when introducing a moderate over-selection margin ($\epsilon = 64$ or 256), validating the robustness of our approach across different frontier DSA-based models. Selected RULER results and the finer $\epsilon = 32$ GLM-5 ablation are in Appendix B; they include task-dependent gains and regressions.

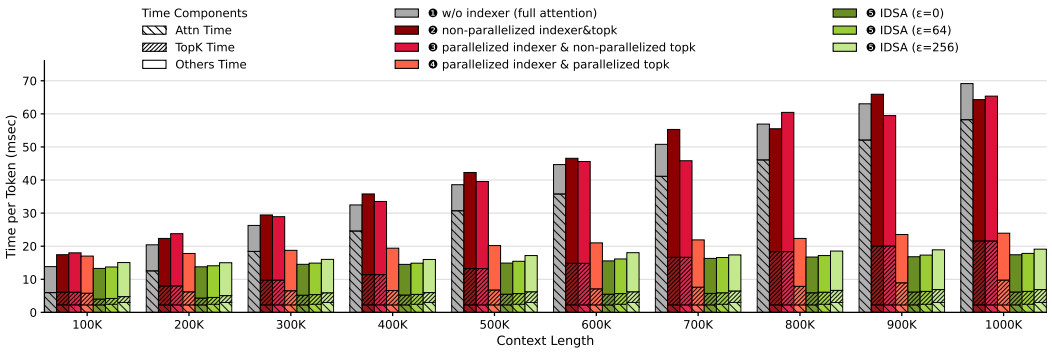

Figure 4: Time per Token Breakdown (msec) v.s. Context Length under different Indexer deployment policies. The execution time is decomposed into Attn Time, TopK Time, and Others Time.

Scheme	DSA & Barrier Cost
❶	$\mathcal{A}ttn(N)$
❷	$\mathcal{I}(N) + \mathcal{T}(N \log K) + \mathcal{A}ttn(K)$
❸	$\mathcal{I}(\frac{N}{G}) + \mathcal{B}(N) + \mathcal{T}(N \log K) + \mathcal{A}ttn(K)$
❹	$\mathcal{I}(\frac{N}{G}) + \mathcal{T}(\frac{N}{G} \log K) + \mathcal{B}(G \cdot K) + \mathcal{T}(G \cdot K \log K) + \mathcal{A}ttn(K)$
❺	$\mathcal{I}(\frac{N}{G}) + \mathcal{T}(\frac{N}{G} \log(\frac{K}{G} + \epsilon)) + \mathcal{B}(K + \epsilon \cdot G) + \mathcal{A}ttn(K + \epsilon \cdot G)$

Table 6: Complexity across attention schemes. N: Context Size; K: Selection Size; G: Number of GPUs; ϵ: per-GPU redundancy; $\mathcal{B}(x)$: inter-GPU barrier cost; $\mathcal{I}(x)$: per-GPU indexer cost; $\mathcal{T}(x)$: per-GPU Top-K cost; $\mathcal{A}ttn(x)$: attention cost per-query-token.

4.3 Efficiency

To better understand the performance implications of different indexer and Top-K execution strategies during the decoding phase, we analyze several inference configurations that vary in how the Lightning Indexer and the Top-K operator are parallelized across GPUs. Both DeepSeek-V3.2 and GLM-5 use DSA-style token-level sparse attention and support the same local-selection principle. We use DeepSeek-V3.2 to evaluate decoding efficiency (Figure 4), and GLM-5 to evaluate prefill efficiency in both single-turn and multi-turn dialogue settings, covering fresh-session and incremental prefill, respectively (Appendix C). All performance experiments were conducted on an 8-card AMD MI300 machine (192GB HBM3 per card) with XGMI high-speed interconnect, using the Intel(R) Xeon(R) Platinum 8480C processor, running model inference on our inference platform (Appendix F). We compare five execution schemes using illustrative cost models in Table 6 and report their empirical per-token decoding time, decomposed into Attention, TopK, and Others (including indexer computation and barrier overheads), with scaling trends from 100K to 1M contexts. The per-token decoding times in Figure 4 cover the complete model forward pass across all layers, including MoE computation and inter-GPU communication, and exclude prefill.

❶ Full attention without indexer. The Lightning Indexer is entirely disabled and the model falls back to full attention computation. As shown in Table 6 (❶), its cost is purely dictated by $\mathcal{A}ttn(N)$. While its latency is competitive at 100K, the attention computation per decoding step scales linearly with context length for fixed head dimensions and counts, since each new query attends to all cached keys and values. As depicted in Figure 4, its execution time per token soars to nearly 70ms at 1000K, which is completely dominated by the *Attn Time* component.

❷ Non-parallelized indexer and Top-K. The Lightning Indexer is enabled, and each GPU executes the full indexer computation and global Top-K selection without parallelism. As indicated by the $\mathcal{I}(N) + \mathcal{T}(N \log K)$ terms in Table 6, this setup fails to exploit multi-GPU parallelism. Consequently, it quickly becomes bottlenecked by the **redundant computation** of full-sequence context-importance scores. In Figure 4, this manifests as massive and rapidly growing *TopK Time* and *Others Time* components, severely degrading efficiency at massive scales.

❸ **Parallelized indexer with non-parallelized Top-K.** Indexer computation is evenly partitioned across $G = 8$ GPUs ($\mathcal{I}(N/G)$). However, the computed scores are then $AllGathered$ to each GPU for a centralized global Top-K selection. This design introduces a massive **communication bottleneck** for gathering full-sequence scores, denoted by $\mathcal{B}(N)$ in Table 6. Empirically, this cross-GPU synchronization severely swells the *Others Time* block, largely offsetting the computational benefits of indexer parallelism and yielding total execution times almost identical to the non-parallelized baseline (❷).

❹ **Two-stage Top-K: local selection followed by global refinement.** Each GPU independently selects its local Top-K candidates from a sequence shard. The resulting G candidate sets are $AllGathered$, followed by a second Top-K on the combined $G \cdot K$ candidates. By reducing the communication payload from the entire sequence $\mathcal{B}(N)$ to just $\mathcal{B}(G \cdot K)$, this hierarchical design significantly improves parallelism, noticeably shrinking the *Others Time* component. However, its reliance on a centralized refinement step $\mathcal{T}(G \cdot K \log K)$ imposes a persistent and visible *TopK Time* overhead, creating a strict performance ceiling.

❺ **Fully distributed indexer and Top-K with IDSA.** IDSA enables a fully distributed configuration where both indexer computation and Top-K selection are partitioned and performed independently. Crucially, it entirely eliminates the second-stage refinement $\mathcal{T}$ and drastically minimizes the barrier cost to $\mathcal{B}(K + \epsilon{\cdot}G)$. By merely requiring careful control of redundancy (ϵ), this design maximizes hardware utilization. Empirically, IDSA effectively compresses both the *TopK* and *Others* time blocks, consistently achieving the lowest overall latency (under 20ms per token even at the extreme 1 million token limit).

Discussion. These configurations expose a clear trade-off between strict global Top-K selection and hardware efficiency. While centralized and multi-stage Top-K schemes (❷, ❸, ❹) attempt to preserve exactness, their theoretical synchronization and computation overheads translate into severely inflated empirical *TopK* and *Others* time delays at long contexts. IDSA (❺) breaks this parallelization barrier. Furthermore, evaluating the stacked bars of IDSA with ϵ = 0, 64, and 256 across Figure 4 reveals that the additional *Attn Time* and *TopK Time* introduced by moderate over-selection is virtually imperceptible (adding less than 1ms combined). This confirms IDSA as a sustainable approach in the evaluated configurations for extreme long-context inference.

4.4 Prefill Efficiency and End-to-End Implications

We complement the decoding analysis with supplementary GLM-5 prefill measurements using 16K-token chunks, covering fresh prefill and extensions of cached prefixes (Appendix C). In the measured baseline (Prefill CP used by vLLM/SGLang), cached keys are gathered before processing the selected context. As the cached prefix grows, this gather transfers more data and incurs greater communication overhead. IDSA instead retains the sharded layout and performs local selection, avoiding this increasingly costly gather of cached keys. Consistent with this communication advantage, IDSA achieves modest MLA-component speedups for fresh prefill and larger speedups for long cached prefixes in the evaluated configurations, relative to the Prefill CP baseline.

5 Limitations and Future Work

While IDSA effectively mitigates the Top-K parallelization bottleneck, several limitations remain and suggest directions for future work.

Theoretical Modeling Gap. Our mathematical upper bound is derived under an independent stochastic assumption based on the Binomial model, which provides a probabilistic full-recall guarantee under independent uniform token assignment. Developing a tighter theoretical model that explicitly captures the intrinsic *local continuity* of attention scores remains an important open problem.

Evaluation Scope and Noise Robustness. We evaluate IDSA on intensive mathematical reasoning (AIME), NIAH, and RULER tasks, with generally comparable scores and task-dependent variation. Future work should further examine its behavior on broader long-context benchmarks (e.g., long-document summarization). In particular, understanding whether moderate over-selection introduces

distracting noise in multi-hop reasoning or information-dense summarization tasks would provide stronger evidence for its general applicability.

Applicability and Architectural Limitations. Our demonstrated scope remains DSA-style token-level sparse attention on the evaluated hardware; other architectures and serving stacks require validation (Appendix D).

6 Conclusion

In this paper, we introduce Interleaved DeepSeek Sparse Attention (IDSA), a distributed approach for scaling fine-grained DSA to million-token contexts. Our key finding is that maintaining strict Top-K exactness during inference can be relaxed. This insight enables us to accelerate cross-GPU context selection through an interleaved context-parallel layout, which reduces cross-GPU dependency and improves parallelization. We provide a recall analysis under randomized assignment and empirically observe generally comparable task quality on DeepSeek-V3.2 and GLM-5, within the stated evaluation scope. Ultimately, IDSA unlocks the parallel efficiency of fine-grained sparse attention in the extreme long-context regime.

Acknowledgments and Disclosure of Funding

This work was conducted during Yifan Guo's research internship at Microsoft Research Asia, which provided internship support and computational resources.

References

[1] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

[2] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[3] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[4] Zihao Yi, Jiarui Ouyang, Zhe Xu, Yuwen Liu, Tianhao Liao, Haohao Luo, and Ying Shen. A survey on recent advances in llm-based multi-turn dialogue systems. *ACM Computing Surveys*, 58(6):1–38, 2025.

[5] Chen Zhang, Xinyi Dai, Yaxiong Wu, Qu Yang, Yasheng Wang, Ruiming Tang, and Yong Liu. A survey on multi-turn interaction capabilities of large language models. *arXiv preprint arXiv:2501.09959*, 2025.

[6] Domenico Ursino, Gianluca Bonifazi, Enrico Corradini, Michele Marchetti, Sarfraz Nawaz Brohi, Qurat ul-ain Mastoi, N. Z. Jhanjhi, and Thulasyammal Ramiah Pillai. A research landscape of agentic ai and large language models: Applications, challenges and future directions. *Algorithms*, 18:499, 2025. URL `https://api.semanticscholar.org/CorpusID:281683215`.

[7] Jiacheng Liu, Xiaohan Zhao, Xinyi Shang, and Zhiqiang Shen. Dive into claude code: The design space of today's and future ai agent systems, 2026. URL `https://arxiv.org/abs/2604.14228`.

[8] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *CoRR*, abs/2107.03374, 2021. URL `https://arxiv.org/abs/2107.03374`.

[9] Jiaheng Liu, Dawei Zhu, Zhiqi Bai, Yancheng He, Huanxuan Liao, Haoran Que, Zekun Wang, Chenchen Zhang, Ge Zhang, Jiebin Zhang, et al. A comprehensive survey on long context language modeling. *arXiv preprint arXiv:2503.17407*, 2025.

[10] Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. Longbench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 3119–3137, 2024.

[11] Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, et al. Longbench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3639–3664, 2025.

[12] MiniMax, Aonian Li, Bangwei Gong, Bo Yang, Boji Shan, Chang Liu, Cheng Zhu, Chunhao Zhang, Congchao Guo, Da Chen, Dong Li, et al. Minimax-01: Scaling foundation models with lightning attention, 2025. URL `https://arxiv.org/abs/2501.08313`.

[13] Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving mamba2 with delta rule, 2025. URL `https://arxiv.org/abs/2412.06464`.

[14] Kimi Team, Yu Zhang, Zongyu Lin, Xingcheng Yao, Jiaxi Hu, Fanqing Meng, Chengyin Liu, Xin Men, Songlin Yang, Zhiyuan Li, Wentao Li, et al. Kimi linear: An expressive, efficient attention architecture, 2025. URL `https://arxiv.org/abs/2510.26692`.

[15] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

[16] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

[17] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks, 2024. URL https://arxiv.org/abs/2309.17453.

[18] DeepSeek-AI, Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, et al. Deepseek-v3.2: Pushing the frontier of open large language models, 2025. URL https://arxiv.org/abs/2512.02556.

[19] Yizhao Gao, Zhichen Zeng, Dayou Du, Shijie Cao, Peiyuan Zhou, Jiaxing Qi, Junjie Lai, Hayden Kwok-Hay So, Ting Cao, Fan Yang, et al. Seerattention: Learning intrinsic sparse attention in your llms. *arXiv preprint arXiv:2410.13276*, 2024.

[20] Yizhao Gao, Shuming Guo, Shijie Cao, Yuqing Xia, Yu Cheng, Lei Wang, Lingxiao Ma, Yutao Sun, Tianzhu Ye, Li Dong, et al. Seerattention-r: Sparse attention adaptation for long reasoning. *arXiv preprint arXiv:2506.08889*, 2025.

[21] Huiqiang Jiang, Yucheng Li, Chengruidong Zhang, Qianhui Wu, Xufang Luo, Surin Ahn, Zhenhua Han, Amir H Abdi, Dongsheng Li, Chin-Yew Lin, et al. Minference 1.0: Accelerating pre-filling for long-context llms via dynamic sparse attention. *Advances in Neural Information Processing Systems*, 37:52481–52515, 2024.

[22] Yucheng Li, Huiqiang Jiang, Chengruidong Zhang, Qianhui Wu, Xufang Luo, Surin Ahn, Amir H Abdi, Dongsheng Li, Jianfeng Gao, Yuqing Yang, et al. Mminference: Accelerating pre-filling for long-context vlms via modality-aware permutation sparse attention. *arXiv preprint arXiv:2504.16083*, 2025.

[23] Jintao Zhang, Chendong Xiang, Haofeng Huang, Jia Wei, Haocheng Xi, Jun Zhu, and Jianfei Chen. Spargeattention: Accurate and training-free sparse attention accelerating any model inference. *arXiv preprint arXiv:2502.18137*, 2025.

[24] Ruyi Xu, Guangxuan Xiao, Haofeng Huang, Junxian Guo, and Song Han. Xattention: Block sparse attention with antidiagonal scoring. *arXiv preprint arXiv:2503.16428*, 2025.

[25] Jingyang Yuan, Huazuo Gao, Damai Dai, Junyu Luo, Liang Zhao, Zhengyan Zhang, Zhenda Xie, Yuxing Wei, Lean Wang, Zhiping Xiao, et al. Native sparse attention: Hardware-aligned and natively trainable sparse attention. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23078–23097, 2025.

[26] Enzhe Lu, Zhejun Jiang, Jingyuan Liu, Yulun Du, Tao Jiang, Chao Hong, Shaowei Liu, Weiran He, Enming Yuan, Yuzhi Wang, et al. Moba: Mixture of block attention for long-context llms. *arXiv preprint arXiv:2502.13189*, 2025.

[27] William Feller. *An introduction to probability theory and its applications.* Wiley & Sons, 1968.

[28] Larry Wasserman. *All of statistics: a concise course in statistical inference*, volume 26. Springer, 2004.

[29] Geoffrey Grimmett and David Stirzaker. *Probability and random processes*. Oxford university press, 2020.

[30] GLM-5 Team, Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, et al. Glm-5: from vibe coding to agentic engineering, 2026. URL https://arxiv.org/abs/2602.15763.

[31] Microsoft. Tutel: An optimized mixture-of-experts implementation. https://github.com/microsoft/Tutel. GitHub repository.

[32] Changho Hwang, Wei Cui, Yifan Xiong, Ziyue Yang, Ze Liu, Han Hu, Zilong Wang, Rafael Salas, Jithin Jose, Prabhat Ram, Joe Chau, Peng Cheng, Fan Yang, Mao Yang, and Yongqiang Xiong. Tutel: Adaptive mixture-of-experts at scale. *CoRR*, abs/2206.03382, 2022. URL https://arxiv.org/abs/2206.03382.

A Supplementary Statistics for Variance Reduction and Recall Bounds

Pattern	4K	8K	16K	32K	64K	128K
Contiguous	500 (244)	982 (726)	1042 (786)	1847 (1591)	1832 (1576)	1704 (1448)
Interleaved	**277** (21)	**284** (28)	**286** (30)	**290** (34)	**289** (33)	**294** (38)
Random	295 (39)	306 (50)	299 (43)	307 (51)	307 (51)	301 (45)

Table 7: Minimum local selection threshold m required for $G = 8$ to guarantee exactly 100% recall of the global Top-K tokens with a 99% probability ($\Pr(\text{Recall} = 100\%) \geq 0.99$). Data is presented in the format $m\,(\epsilon)$, where the over-selection $\epsilon = m - K/G$ and represents the additional tokens retrieved per GPU beyond the ideal uniform baseline of 256.

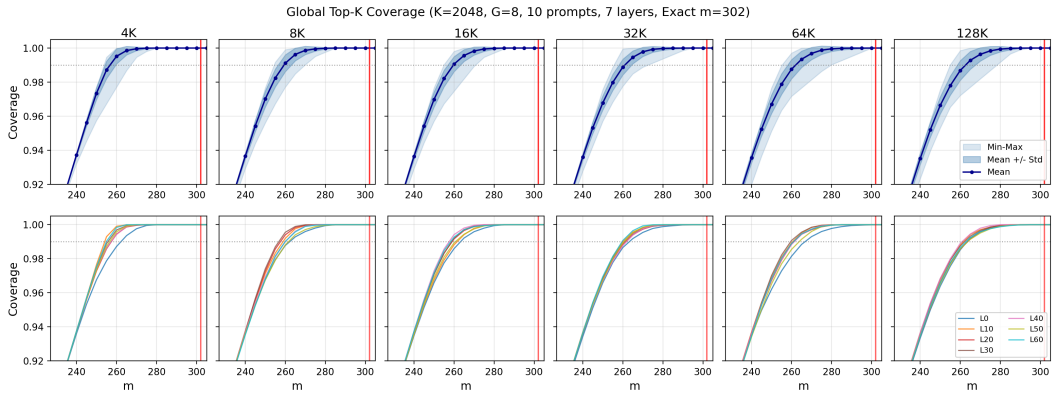

Figure 5: Empirical coverage of the global Top-K tokens under interleaved context parallelism with $G = 8$ GPUs and $K = 2048$. The vertical red line indicates the theoretical binomial upper bound.

B Extended Long-Context Evaluation

We perform the reasoning and retrieval experiments with extended tasks from RULER,[1] evaluated at input lengths of 32K, 64K, and 128K tokens. The task identifiers VT, CWE, NIAH, QA_1, and QA_2 are retained from the evaluation records. These experiments broaden the evidence beyond AIME and the original single-needle retrieval setting, but constitute a subset of RULER rather than a complete long-context evaluation suite.

Tables 8 and 9 report the available scores on DeepSeek-V3.2 and GLM-5, respectively. The reference is the original exact Top-2048 configuration. For IDSA, $K = 2048$ and $G = 8$, so the local budget is $m = 256 + \epsilon$ and the total candidate budget is at most $2048 + 8\epsilon$. In particular, $\epsilon = 32$, 64, and 256 correspond to total budgets of 2304, 2560, and 4096 tokens. DeepSeek-V3.2 was evaluated at temperature 1.0.

For each evaluated context length and configuration, QA_1 and QA_2 each use 50 test cases, while VT and CWE each use 10 test cases. NIAH averages over five variants with 10 cases per variant, for 50 cases in total. Each case is evaluated with one generation attempt; these sample counts do not denote repeated independent runs. The tables report aggregate point estimates from this limited sample, rather than averages over five repeated runs. Without per-example outcomes and uncertainty estimates, small score differences should not be interpreted as statistically significant improvements.

Task	Input length	Exact Top-2048	IDSA ($\epsilon = 32$)	IDSA ($\epsilon = 64$)	IDSA ($\epsilon = 256$)
VT	32K	1.00	1.00	1.00	1.00
	64K	1.00	1.00	1.00	1.00
	128K	1.00	1.00	1.00	1.00
CWE	32K	1.00	1.00	1.00	1.00
	64K	0.92	0.90	0.94	0.98
	128K	0.60	0.60	0.70	0.66
NIAH	32K	1.00	1.00	1.00	1.00
	64K	1.00	0.98	1.00	0.97
	128K	0.97	0.96	0.96	0.98
QA_1	32K	0.82	0.84	0.84	0.90
	64K	0.64	0.62	0.64	0.76
	128K	0.52	0.56	0.58	0.56
QA_2	32K	0.78	0.78	0.80	0.80
	64K	0.80	0.76	0.78	0.82
	128K	0.74	0.74	0.74	0.76

Table 8: Selected RULER results on DeepSeek-V3.2, using the reported official task metrics on a $[0, 1]$ scale (higher is better). Each task and context length is shown separately to retain improvements and regressions.

Task-dependent effects of over-selection. The additional results support useful, but non-uniform, robustness. On DeepSeek-V3.2, QA_1 at 128K improves from 0.52 to 0.58 with $\epsilon = 64$, while NIAH at the same length changes from 0.97 to 0.96. On GLM-5, NIAH at 128K changes from 0.92 to 1.00 with $\epsilon = 64$, but reaches 0.93 with the larger $\epsilon = 256$ budget. GLM-5 QA_1 at 64K decreases from 0.88 to 0.84 at $\epsilon = 32$, returns to 0.88 at $\epsilon = 64$, and reaches 0.86 at $\epsilon = 256$. Thus, increasing the budget does not imply monotonic task accuracy, and neither full recall of the reference set nor admitting additional tokens establishes that distracting context is harmless.

Scope of the evidence. The $\epsilon = 32$ results provide a finer budget ablation for GLM-5, but do not isolate the causal effect of recall: budget, selected tokens, and attention normalization change together. The available records do not contain a matched per-query recall-versus-accuracy ablation or corresponding $\epsilon = 32$ AIME results. Long-document summarization, code understanding, a complete LongBench evaluation, and comprehensive multi-hop reasoning remain outside the demonstrated

[1] Cheng-Ping Hsieh et al., *RULER: What's the Real Context Size of Your Long-Context Language Models?*, 2024. `https://arxiv.org/abs/2404.06654`.

Task	Input length	Exact Top-2048	IDSA $\epsilon = 32$	IDSA $\epsilon = 64$	IDSA $\epsilon = 256$	Full attention
VT	32K	1.00	1.00	1.00	1.00	1.00
	64K	1.00	1.00	1.00	1.00	1.00
	128K	1.00	1.00	1.00	1.00	1.00
NIAH	32K	1.00	1.00	1.00	0.99	1.00
	64K	0.97	0.96	0.96	0.97	0.97
	128K	0.92	0.92	1.00	0.93	1.00
QA_1	32K	0.90	0.88	0.90	0.88	0.92
	64K	0.88	0.84	0.88	0.86	0.90
	128K	0.86	0.86	0.86	0.90	0.94
QA_2	32K	0.80	0.80	0.80	0.80	0.80
	64K	0.80	0.80	0.80	0.80	0.80
	128K	0.62	0.60	0.62	0.62	0.66
CWE	32K, 64K, 128K	Not scored: context limit exceeded				

Table 9: Selected RULER results on GLM-5, including the finer $\epsilon = 32$ setting and full attention. CWE could not be scored because repeated self-reflection exhausted the reported 200K context limit before a final answer; this behavior also occurred with the exact Top-2048 baseline. These missing results are not zeros and are not evidence of preserved accuracy.

scope. In particular, the missing GLM-5 CWE scores and the observed QA variation should be retained when interpreting the robustness of IDSA.

C Prefill Measurements and End-to-End Interpretation

Measurement scope. We supplement the decoding analysis with GLM-5 prefill measurements using 16K-token chunks and a total extension of 128K tokens. The starting cached contexts are 0, 512K, and 896K tokens, producing the three ranges in Table 10. The latter two settings are systems scaling tests beyond the model context range used for the quality evaluation; they do not establish reliable reasoning or retrieval at 640K or 1024K tokens. The table reports MLA component times and their percentages of full-model prefill latency over the same measurement interval. Here, End-to-End refers to complete prefill, including both MLA and non-MLA components, and excludes decoding.

Configuration	Fresh prefill 0–128K	Cached extension 512–640K	Cached extension 896–1024K
IDSA ($\epsilon = 0$)	4.27 (28.0%)	5.26 (32.4%)	6.24 (36.2%)
IDSA ($\epsilon = 32$)	4.34 (28.3%)	5.26 (32.4%)	6.24 (36.2%)
IDSA ($\epsilon = 64$)	4.34 (28.3%)	5.27 (32.4%)	6.25 (36.2%)
IDSA ($\epsilon = 256$)	4.51 (29.1%)	5.61 (33.8%)	6.44 (36.9%)
Prefill CP	4.76 (30.2%)	7.24 (39.7%)	10.85 (49.7%)

Table 10: Supplementary GLM-5 prefill measurements. Entries are reported MLA component times in seconds, followed by their reported E2E shares. Prefill CP is the measured reference context-parallel policy. These component measurements and percentages are not a separately measured prompt-plus-generation latency benchmark.

Where the prefill benefit appears. The measured Prefill CP reference gathers cached keys before processing the selected context, whereas IDSA retains the sharded layout and applies local selection. Avoiding a repeated gather of an increasingly large cached prefix can benefit multi-turn extensions. For example, at $\epsilon = 256$, the tabulated MLA speedups are $4.76/4.51 \approx 1.06\times$ for fresh prefill, $7.24/5.61 \approx 1.29\times$ for the 512K cached prefix, and $10.85/6.44 \approx 1.68\times$ for the 896K cached prefix. At $\epsilon = 64$, the corresponding ratios are approximately $1.10\times$, $1.37\times$, and $1.74\times$. These are ratios of the measured component times. Other model components, including the MoE layers, limit the resulting overall gain. The cache-communication saving does not imply that total prefill computation is independent of cached context length.

Let A denote the MLA component time and r its percentage of full-model prefill latency. The corresponding total prefill latency is $P = A/(r/100)$. Using the rounded values in Table 10, the prefill E2E speedups at $\epsilon = 256$ are approximately $1.02\times$, $1.10\times$, and $1.25\times$ for fresh prefill and the two cached-extension settings, respectively.

When faster decoding improves a complete request. Let P_B and P_I be full-model prefill latency for the baseline and IDSA. For an input of N tokens and a fixed output length L, let $d_B(n)$ and $d_I(n)$ denote full-model per-token decoding latency at context length n. Excluding queueing and external tool execution, the model-execution speedup is

$$S_{\text{request}} = \frac{P_B + \sum_{j=1}^{L} d_B(N + j - 1)}{P_I + \sum_{j=1}^{L} d_I(N + j - 1)}. \tag{10}$$

Writing $D_B = \sum_j d_B(N + j - 1)$, $D_I = \sum_j d_I(N + j - 1)$, $f = P_B/(P_B + D_B)$, $s_P = P_B/P_I$, and $s_D = D_B/D_I$ gives

$$S_{\text{request}} = \left(\frac{f}{s_P} + \frac{1 - f}{s_D} \right)^{-1}. \tag{11}$$

If prefill is unchanged ($s_P = 1$), even arbitrarily fast decoding cannot exceed $1/f$ speedup for $f > 0$. Consequently, a large decoding speedup need not yield a large request-level speedup on long-input, short-output workloads. Conversely, long reasoning traces or repeated generations over a reused prefix can make D_B exceed P_B, increasing the impact of decoding improvements. With approximately constant baseline decode latency $\bar{d}_B$, decoding dominates only when $L\bar{d}_B > P_B$; a context length exceeding 100K alone does not imply this condition. End-to-end gains therefore depend on input and output lengths, prefix reuse, and prefill implementation. Serving throughput and tail latency additionally require measurements under the intended batching and scheduling policy.

D Applicability and Architectural Limitations

What is demonstrated. The quality evidence concerns the DSA-style token-level selection used by DeepSeek-V3.2 and GLM-5. Evaluating two models within this family is useful, but is not experimental evidence for arbitrary sparse-attention architectures. The main decoding measurements use the reported eight-GPU MI300/XGMI setup; the supplementary prefill results concern the measured GLM-5 policies. These results do not establish equivalent gains on other accelerators, inter-node networks, or production serving stacks.

Conditions for transferring the selection rule. The partition-and-union construction can be considered when a model exposes comparable per-candidate importance scores, a reference global Top-K set, and a disjoint partition of the candidates. Each partition must evaluate the same score ordering, with consistent tie-breaking, and downstream attention must accept the union of local selections without enforcing a second exact global budget. The backbone must also tolerate the resulting candidate set. The choice of MLA, MHA, or GQA alone does not establish these properties; head-specific scoring and KV sharing may require different layouts and kernels. Thus, transferring the selection rule is a design possibility subject to quality and systems validation, rather than a demonstrated drop-in replacement.

Architectures requiring additional work. Block-sparse methods select groups of tokens and often rely on contiguous blocks for efficient memory access. Applying token-level interleaving can disrupt that structure; an extension would need block-level partitioning and budget analysis, followed by kernel and quality evaluation. Fixed-window or fixed-pattern attention has no equivalent global content-based Top-K bottleneck, while recurrent or linear-attention formulations need not expose an explicit retrievable KV candidate set. IDSA therefore does not directly provide the same optimization for these mechanisms. Even within token-level sparse attention, periodic or strongly imbalanced importance patterns can require a much larger local budget, reducing or eliminating the benefit.

Serving integration and remaining communication. An implementation must preserve token positions and positional encodings, support causal masks and paged KV-cache access, and combine shard-local attention results with correct global softmax normalization. Removing the second global Top-K refinement does not remove all cross-device communication: query distribution, attention-output reductions, and model-parallel collectives may remain. The measured Prefill CP policy illustrates one communication trade-off; it should not be read as a benchmark of every version or configuration of vLLM or SGLang. Likewise, extensions to CPU threads or single-GPU multiprocessors require separate evaluation because their synchronization and memory costs differ from the reported multi-GPU setting.

E Recall Guarantees, Interleaving, and Model Quality

Exact selection versus retaining the reference set. Fix a query and layer, and let T be the reference global Top-K set under the indexer scores. Let $S = \bigcup_{g=0}^{G-1} \mathrm{Top}(\mathcal{P}_g, m)$ be the union of local selections. IDSA relaxes the requirement $S = T$, allowing $|S|$ to exceed K. The recall objective is instead $T \subseteq S$. These statements are compatible: over-selection can preserve all reference candidates without performing exact global pruning. High recall is a safeguard against losing candidates preferred by the pretrained indexer, not a claim that every Top-K token is indispensable or that exact agreement with the indexer guarantees task accuracy.

For disjoint partitions and a consistent total ordering of scores, define $X_g = |T \cap \mathcal{P}_g|$. All reference tokens in a partition rank above its non-reference tokens, so

$$\mathrm{Recall}(T, S) = \frac{1}{K} \sum_{g=0}^{G-1} \min\{X_g, m\}, \qquad T \subseteq S \iff \max_g X_g \le m. \tag{12}$$

The probability of full recall, $\Pr(\mathrm{Recall} = 1)$, is distinct from expected token recall, $\mathbb{E}[\mathrm{Recall}]$. A target such as $\Pr(\mathrm{Recall} = 1) \ge 0.99$ is neither a 99% task-accuracy guarantee nor a guarantee that all layers and all generated tokens in a request simultaneously retain their reference sets.

Scope of the stochastic theorem. Theorem 3.1 applies when each reference token is assigned independently and uniformly to one of G partitions, independently of its score. Conditional on the fixed reference set, $X_g \sim \mathrm{Binomial}(K, 1/G)$, and the union bound controls the chance that any partition contains more than m reference tokens. The X_g are not independent across partitions because they sum to K; their independence is not needed by the union bound. A random permutation into exactly equal-sized shards is a different sampling model, with hypergeometric rather than binomial marginals. The theorem should therefore be interpreted according to its explicit independent-assignment assumption. The normal approximation is an engineering estimate; a stated probability target should be checked using the exact distribution.

Deterministic interleaving assigns position t to $t \bmod G$. To transfer the binomial tail bound to this layout, one would need an additional assumption or proof controlling $\Pr(X_g^{\mathrm{Int}} > m)$, with probability taken over a specified distribution of queries and score patterns. For example, per-partition tail domination by the binomial model would suffice. Lower empirical variance or positive local autocorrelation alone does not establish such tail domination.

A deterministic bound using contiguous high-score regions. A separate, explicit condition connects interleaving to local continuity. Suppose the reference positions T are the union of R disjoint contiguous intervals with lengths $\ell_1, \ldots, \ell_R$, where $\sum_{r=1}^{R} \ell_r = K$. Under round-robin assignment, any interval contributes at most $\lceil \ell_r / G \rceil$ tokens to any partition. Consequently,

$$\max_g X_g^{\mathrm{Int}} \le B(T, G) := \sum_{r=1}^{R} \left\lceil \frac{\ell_r}{G} \right\rceil \le \frac{K}{G} + \frac{G-1}{G} R. \tag{13}$$

Thus, $m \ge B(T, G)$ is sufficient for full reference-set recall for that query and layer. The first inequality follows by counting residues modulo G within each interval and summing across intervals. The second follows from $\lceil \ell/G \rceil \le \ell/G + (G-1)/G$ for positive integer ℓ. If every interval length is divisible by G, the load is exactly K/G in every partition. This is a conditional deterministic statement about the Top-K indicator pattern, not a consequence of visually smooth scores alone. The current experiments do not report the interval counts or lengths needed to certify $m = 320$ using this bound.

The condition also makes the limitation concrete. If all K reference positions have the same residue modulo G, one partition contains all of them and full recall requires $m \ge K$. Interleaving therefore has no distribution-free near-K/G guarantee. Moreover, with fixed K and G, the independent-assignment variance $K(1/G)(1 - 1/G)$ does not depend on the total context length N. An empirical narrowing of the gap between layouts as N grows cannot be attributed to a law-of-large-numbers argument in N under this fixed-K model; it requires evidence about how the score patterns change.

Why high recall helps, and why it is not sufficient. Missing reference tokens can remove information that the pretrained selector ranks highly. In Table 5, GLM-5 changes from 140/139/141 at $\epsilon = 0$ to 141/140/143 at $\epsilon = 64$ on AIME-24/25/26. This is consistent with a useful over-selection margin, but does not isolate recall from changes in budget and candidate composition or prove that a particular recall threshold is necessary. The RULER results in Appendix B further show that larger budgets can improve some tasks and degrade others.

Even when $T \subseteq S$, additional tokens change attention normalization. To illustrate this at one attention operation, keep the query, keys, values, and attention logits fixed. Write $S = T \cup E$ with $E \cap T = \varnothing$, and let a_t be softmax weights normalized over S. If $\eta = \sum_{t \in E} a_t$ and $\|v_t\| \leq V$, then the attention outputs satisfy

$$o_S = (1 - \eta)o_T + \eta o_E, \qquad \|o_S - o_T\| \leq 2V\eta, \tag{14}$$

where o_T and o_E use weights renormalized on their respective sets; when E is empty the difference is zero. The inequality follows because both outputs are convex combinations of values with norm at most V. It shows why the relevant quantity is the additional attention mass, not just the number of extra tokens. Indexer ranks are proxies for attention importance and do not by themselves bound η. Likewise, omitting a token with substantial attention mass can matter despite high count-based recall. This single-operation calculation does not provide a bound on autoregressive task accuracy, where subsequent hidden states and generated tokens may diverge.

Practical interpretation of the budget. For the evaluated eight-way configuration, $\epsilon = 32$ gives $m = 288$, while $\epsilon = 64$ gives the hardware-aligned $m = 320$. The former probes a smaller budget near the empirical recall thresholds; the latter provides additional margin relative to the stochastic reference calculation and the observed interleaved thresholds. Neither is a universal setting for other models, workloads, or partition counts. The evidence supports the narrower conclusion that IDSA can preserve high reference-set recall and competitive task scores with modest over-selection on the evaluated DSA workloads, while the guarantee for randomized assignment and the conditional bound in Eq. (13) retain their stated assumptions.

F Code Availability and Reproducibility

We describe the inference service, the evaluation entry point, and the source-level implementation of IDSA below.

F.1 Inference service

The inference service is distributed through Docker Hub at `https://hub.docker.com/r/tutelgroup/deepseek-671b`. The following command targets a Linux Docker/ROCm host with eight AMD MI300X GPUs and exposes the service on port 8000. It uses `nvidia/GLM-5.3-NVFP4` as a example of latest IDSA-supported model.

The shell variable `EPS` specifies the per-GPU over-selection margin ϵ and is used to construct the total candidate budget `TOP` $= K + G\epsilon$. With $K = 2048$, $G = 8$, and $\epsilon = 64$, the total budget is 2560 and the local budget is $m = 320$. This matches the budget of IDSA-Top2560 in Table 5, but denotes local Top-320 selection on each interleaved shard, not exact global Top-2560 selection.

```
$ export EPS=64

$ docker run --name idsa_serve -e TOP=$((2048 + 8 * EPS)) \
  -e WORKER=32 -e LOCAL_SIZE=8 -p 8000:8000 -it --rm --ipc=host --shm-size=8g \
  --cap-add=SYS_PTRACE --security-opt seccomp=unconfined \
  --ulimit memlock=-1 --ulimit stack=67108864 -v /:/host -w /host$(pwd) \
  --device=/dev/kfd --device=/dev/dri --group-add=video \
  tutelgroup/deepseek-671b:mi300x8-chat-20261020 \
  --serve=core --try_path nvidia/GLM-5.3-NVFP4 \
  --max_seq_len 200000 --thinking_effort max
```

This launch configuration mounts the host filesystem and relaxes container isolation; it should only be used with a trusted image on a trusted, dedicated host.

F.2 Evaluation client

The evaluation script can be downloaded in the Tutel repository [31, 32] following the instructions below. Start the inference service before invoking the AIME-2025 evaluation entry point:

```
$ docker exec -it idsa_serve bash
$ curl -LO https://raw.githubusercontent.com/microsoft/Tutel/refs/heads/main/tests/test_idsa.py

$ DS=aime-2025 python test_idsa.py
```

This example documents the evaluation interface for the service release above. Reproducing the paper's reported scores additionally requires the corresponding model and precision, decoding settings, task data, and run counts; the example command alone does not specify all of these experimental conditions.

F.3 IDSA source

The image includes the source file `/opt/deepseek-tutel-accel/model_deepseek_moe.py`, where the core local-selection step is implemented in `deepseek_indexer`. IDSA partitions indexer computation and candidate selection across interleaved context shards. The selected candidates can subsequently be consumed through either candidate gathering or shard-local attention.

In the distributed selection step, `index_topk` represents the total over-selected budget, and `index_topk // world_size` determines the per-GPU output capacity of `topk_indices_out`. For $K = 2048$, $G = 8$, and $\epsilon = 64$, the local budget is $m = K/G + \epsilon = 320$. This denotes independent local Top-320 selection, rather than exact global Top-2560 selection.

The metadata kernel `kv_build_metadata` computes the valid local context length for each sequence and rank. These lengths are exposed as `token_range.seq_lens_cp` and passed to the local selection operator to delimit its search range. Thus, the local context length specifies where to search, whereas the output capacity specifies how many candidates to select. All ranks execute this local-selection step in parallel.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly outline the claims about exact Top-K necessity, the proposed IDSA method, and throughput improvements, which are consistently supported by theoretical analysis in Section 3.4 and experiments in Section 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 5 discusses current limitations regarding the theoretical modeling gap, evaluation scope, and architectural limitations.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Section 3.4 explicitly states the assumption of independent uniform distribution for the worst-case bound and derives the formal thresholds (Theorem 3.1 and Equation (7)) with step-by-step proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper details the models used (DeepSeek-V3.2, GLM-5), sequence lengths, Top-K settings, layout configurations, hardware environment, and benchmark parameters in Section 4.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Appendix F provides access to the IDSA implementation source distributed in the Docker image and the evaluation scripts hosted in the Tutel repository.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Since this is a train-free inference algorithm, all test details such as test datasets (AIME, NIAH), inference temperatures, hardware configurations, and evaluation modes are explicitly specified in Section 4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Table 2 reports standard deviations for spatial load variance. To ensure reliability and mitigate randomness, the AIME benchmark results (Table 1 and Table 5) and the Needle-In-A-Haystack results (Figure 2) are aggregated over 5 and 10 independent runs, respectively.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section 4.3 explicitly documents the hardware details, specifying the use of an 8-card AMD MI300 machine (192GB HBM3 per card) with an Intel Xeon Platinum 8480C processor.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research focuses on fundamental algorithmic and system efficiency for LLM inference and conforms entirely to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [N/A]

Justification: This work focuses on fundamental hardware-efficient algorithms for existing large language models and does not introduce new societal impacts beyond those already associated with deploying LLMs.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release new pre-trained models or scraped datasets; it proposes an inference-time algorithm applied to existing open-weights models.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper appropriately cites and credits the creators of the existing models (DeepSeek-V3.2, GLM-5) and benchmark tasks utilized in the experiments.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper introduces a novel algorithmic method rather than new datasets or model assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The research does not involve crowdsourcing or human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

Justification: The research does not involve human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method development in this research does not incorporate LLMs as components of the proposed approach. Instead, LLMs are treated purely as the subject of evaluation and optimization, and are not used as tools, modules, or integral parts in the methodological design.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.