MISA: MIXTURE OF INDEXER SPARSE ATTENTION FOR LONG-CONTEXT LLM INFERENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

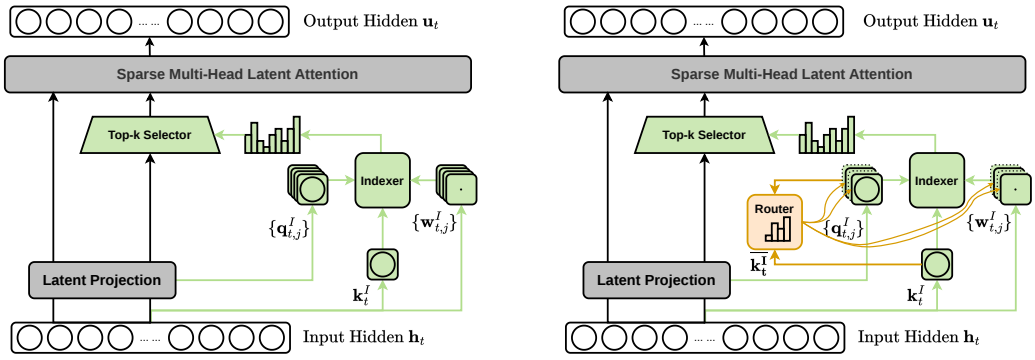
DeepSeek Sparse Attention (DSA) uses a learned multi-head indexer to score prefix tokens and select the top-k tokens for sparse attention, but indexing becomes costly at long context lengths. We propose **MISA** (**M**ixture of **I**ndexer **S**parse **A**ttention), a training-free, drop-in replacement that treats indexer heads as a pool of experts. A lightweight router uses block-level statistics to select a query-dependent subset of $h \ll H^I$ active heads for token-level scoring, reducing the per-query indexing cost from $\mathcal{O}(H^I L)$ to $\mathcal{O}(hL + H^I M)$, where $M = \lceil L/B \rceil$. A hierarchical variant, MISA†, further re-ranks an enlarged candidate set with the original DSA indexer. On the 13-subtask RULER benchmark across six context lengths from 4K to 128K, MISA† achieves an average score of 94.94, second only to DSA at 95.62. End-to-end prefill evaluation shows that MISA reduces mean time to first token by a factor of $1.42\times$ on the full model at 256K and $1.57\times$ on a pruned 8-layer model at 512K. These results demonstrate that head-axis routing provides a practical quality–efficiency trade-off for fine-grained sparse attention.

1 INTRODUCTION

Frontier large language models such as Qwen3 (Yang et al., 2025), Kimi K2 (Moonshot AI, 2025), GLM-5 (Zeng et al., 2026), and DeepSeek-V3.2 (DeepSeek-AI, 2025) now routinely process prefixes of hundreds of thousands of tokens in a single forward pass, and the latest releases—GPT-5.5 (OpenAI, 2026), Claude Opus 4.7 (Anthropic, 2026), Gemini 3 (Google DeepMind, 2026), MiniMax-01 (Li et al., 2025), and DeepSeek-V4 (DeepSeek-AI, 2026)—push this to the million-token regime. At these lengths, dense attention becomes the dominant cost of both prefill and decode, motivating a wave of *sparse attention* techniques that select only a small subset $\mathcal{T}_t$ of past tokens for each query position (Child et al., 2019; Beltagy et al., 2020; Zaheer et al., 2020; Xiao et al., 2024b; Zhang et al., 2023; Li et al., 2024; Tang et al., 2024; Xiao et al., 2024a; Chen et al., 2024; Gao et al., 2024; Lu et al., 2025; Yuan et al., 2025; Xu et al., 2026). Among them, DeepSeek Sparse Attention (DSA) (DeepSeek-AI, 2025) stands out as the best-performing fine-grained variant in production: rather than selecting whole blocks of tokens with handcrafted patterns, DSA introduces a lightweight learned *indexer* that scores every prefix token and feeds the top-k tokens into the main attention. Its dominance carries into the next generation: DeepSeek-V4's Compressed Sparse Attention (CSA) (DeepSeek-AI, 2026) is, at its core, DSA applied on top of a $4\times$ compressed (block-level) KV stream, confirming that learned token-wise indexing remains the strongest building block even when the underlying KV is itself compressed. By contrast, block-level inference-time alternatives such as MoBA (Lu et al., 2025) consistently lag behind DSA on retrieval-style benchmarks because their per-block scores cannot localise the relevant content within a block.

A central design choice of DSA is that the indexer itself is multi-head: although the main attention in DeepSeek-V3.2 has $H = 128$ query heads, all of them share the *same* selected token set $\mathcal{T}_t$ (the sparse MLA operates in MQA mode with a single key/value entry per token), so a single indexer is sufficient—yet DSA still uses $H^I = 64$ indexer heads. The reason is expressiveness: each head specialises in a different relevance pattern (recency, syntactic role, lexical or semantic similarity, ...), and the aggregated score $I_{t,s} = \sum_j w^I_{t,j}\,\mathrm{ReLU}(\mathbf{q}^I_{t,j} \cdot \mathbf{k}^I_s)$ benefits from this diversity, with measurable retrieval degradation as H^I shrinks. The unfortunate consequence is that scoring each of the L prefix tokens with all H^I heads is precisely what makes the indexer the dominant cost on long contexts. Two recent improvements try to reduce this indexer cost without touching the head

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) DSA (b) MISA

Figure 1: Comparison of the DSA and MISA indexers. **(a) DSA** scores every prefix token with all H^I indexer heads in parallel before the Top-k Selector picks the final tokens. **(b) MISA** introduces a lightweight *Router* that uses block-pooled indexing keys $\tilde{\mathbf{k}}^I_b$ (cf. Eq. 4) to choose a query-dependent subset of $h \ll H^I$ active heads, and only those heads compute the per-token score. Both designs feed the same Sparse Multi-Head Latent Attention operator and produce a top-k token set of identical size, so MISA is a drop-in replacement for the DSA indexer.

axis. IndexCache (Bai et al., 2026) retains only a small set of layers as *full* indexer layers and lets the remaining *shared* layers reuse their top-k, amortising the cost *across layers*; HISA (Xu et al., 2026) attacks the cost from the *token* axis with a hierarchical block-to-token search. Both still keep every one of the H^I heads active inside the kernel, and both step away from DSA's strict per-token, per-layer scoring—tokens outside HISA's selected blocks and entire shared layers in IndexCache no longer receive a fresh fine-grained score, sacrificing part of the granularity that makes DSA strong.

This paper takes the orthogonal view that the bottleneck is the *head* axis. Our key observation is that, while diversity across heads is essential when aggregated over a large pool, only a few heads are actually informative for any given query: the relevant set changes slowly along the prefix and can be identified from cheap block-level statistics. We turn this observation into MISA (**M**ixture of **I**ndexer **S**parse **A**ttention), an indexer that treats the H^I heads as a pool of MoE experts, routes a query-dependent subset of $h \ll H^I$ active heads via a block-pooled scorer, and runs the heavy token-level scan with only those heads. The router itself operates on $M = \lceil L/B \rceil \ll L$ pooled keys, so its overhead is negligible, and the per-query indexer cost is reduced from $\mathcal{O}(H^I L)$ to $\mathcal{O}(hL + H^I M)$. MISA preserves the full diversity of the indexer pool because every head remains available; routing simply chooses *which* ones to consult on each token. We further show that MISA can be plugged into a coarse-to-fine pipeline (MISA†) that uses an enlarged routed candidate set followed by a token-level DSA refinement, improving agreement with the dense indexer's selected token set.

Contributions. (i) We identify the indexer's head–token scoring as a major bottleneck in long-context DSA inference and introduce head-axis routing as an efficiency axis complementary to token-axis hierarchies and cross-layer index reuse. (ii) We propose MISA, a training-free, drop-in indexer that routes a small subset of heads for token-level scoring, together with a hierarchical variant, MISA†, that refines routed candidates using the original DSA indexer. (iii) On RULER, MISA† achieves the second-highest average score among the evaluated methods, trailing DSA by only 0.52 points. It preserves full Needle-in-a-Haystack accuracy up to 128K context. (iv) We demonstrate end-to-end prefill speedups measured by mean time to first token: MISA achieves $1.42\times$ on the full model at 256K and $1.57\times$ on a pruned 8-layer model at 512K. Component profiling and concurrency experiments further characterize the source and robustness of these gains.

2 RELATED WORK

Sparse attention. A long line of work attacks the quadratic cost of attention on long contexts by selecting a subset of past tokens for each query. *Static-pattern* methods such as Sparse Transformer (Child et al., 2019), Longformer (Beltagy et al., 2020), and BigBird (Zaheer et al., 2020) use predefined window, stride, and global tokens that are decoupled from the actual content. *Cache-eviction* methods drop tokens at decode time using attention-statistics heuristics: StreamingLLM (Xiao et al., 2024b) keeps a few attention sinks plus a recent window, H2O (Zhang et al., 2023) retains heavy hitters in past attention, and SnapKV (Li et al., 2024) clusters and compresses the KV cache. These methods avoid retrieval entirely and can therefore lose information that becomes relevant later in the generation.

A more recent class of methods uses *content-based dynamic retrieval* at the block level. Quest (Tang et al., 2024) and InfLLM (Xiao et al., 2024a) score blocks of tokens by their pooled key against the query and select the top-m blocks at decode time. MoBA (Lu et al., 2025) extends this idea to a trained inference-time selector with the same block-level granularity. MagicPIG (Chen et al., 2024) replaces top-k retrieval with LSH-based importance sampling. SeerAttention (Gao et al., 2024) learns a sparse gate jointly with the model. Native Sparse Attention (Yuan et al., 2025) is the first work to train a transformer with a hardware-aligned sparsity pattern from scratch. Among inference-time methods, DeepSeek Sparse Attention (DeepSeek-AI, 2025) is currently the strongest: a learned token-wise indexer scores every prefix token (rather than every block), and the resulting top-k matches dense attention quality at production scale. HISA (Xu et al., 2026) accelerates the DSA indexer with a coarse-to-fine block-to-token search, and IndexCache (Bai et al., 2026) amortises the same indexer across consecutive layers by retaining only a few *full* indexer layers and reusing their top-k in the rest; both still keep every indexer head active inside the kernel and depart from per-token, per-layer scoring. MISA is complementary to all of these works: instead of acting along the token axis, we route along the head axis of the indexer itself, which can be combined orthogonally with any token-level scheme.

Mixture of experts in language models. Conditional computation via mixture of experts (MoE) was introduced by Shazeer et al. (2017) and scaled up in GShard (Lepikhin et al., 2020) and Switch Transformer (Fedus et al., 2022), where a learned router activates a sparse subset of expert FFNs per token. Modern open-weight LLMs such as Mixtral (Jiang et al., 2024) and DeepSeek-MoE (Dai et al., 2024) make their FFN layers MoE-based while leaving the attention layers dense. *Attention-side* MoE has also been explored: Mixture-of-Attention-Heads (Zhang et al., 2022) treats whole multi-head attention modules as experts and routes one module per token; MoH (Jin et al., 2024) treats each individual attention head as an expert and selects a sparse subset of heads to compute the attention output. MISA adopts the same head-as-expert philosophy as MoH but applies it to a fundamentally different module. MoH and prior attention MoEs route the heads that produce the attention *output*, so changing the routed subset directly changes the values written back into the residual stream. MISA instead routes the heads that produce the indexer *score*: the routed subset only decides which similarity patterns are consulted when picking the top-k tokens, while the downstream attention itself remains dense over the chosen set. This decoupling is what makes it possible to use a very small number of experts ($h \ll H^I$) without harming model quality.

3 PRELIMINARIES

We briefly review DeepSeek Sparse Attention (DSA) as used in DeepSeek-V3.2 (DeepSeek-AI, 2025) and a follow-up indexer-side improvement, HISA (Xu et al., 2026); both serve as baselines and as the starting points for the design of MISA. DSA consists of two components: a **token-wise indexer** that selects a small set of relevant tokens for each query, and **Sparse MLA** that performs attention only over the selected tokens. HISA introduces a **hierarchical indexing** mechanism that improves indexer efficiency while keeping Sparse MLA identical to DSA. The notation introduced below is used throughout the rest of the paper.

Indexer in DSA. Let L denote the causal prefix length for a query position t. The indexer maintains lightweight indexing keys $\mathbf{k}_s^I$, indexing queries $\mathbf{q}_{t,j}^I$ for H^I indexing heads, and per-head gating

weights $w^I_{t,j}$. The relevance score between query t and key s is

$$I_{t,s} = \sum_{j=1}^{H^I} w^I_{t,j} \cdot \mathrm{ReLU}\big(\mathbf{q}^I_{t,j} \cdot \mathbf{k}^I_s\big)\,. \tag{1}$$

The indexer then selects the top-k token indices,

$$\mathcal{T}_t = \mathrm{TopK}(I_{t,:},\, k), \tag{2}$$

which are passed to the downstream Sparse MLA operator. The per-query indexer cost is therefore $\mathcal{O}(H^I L)$, dominated by the H^I head–token products in Eq. 1; aggregated over a full prefill it grows as $\mathcal{O}(H^I L^2)$ per layer.

Sparse MLA in DSA. Sparse MLA adopts the MQA mode of MLA, in which each token stores a single latent key–value entry $\mathbf{c}_s$ shared across all query heads. Given the selected token set $\mathcal{T}_t$, attention is computed only over the selected entries:

$$\mathbf{u}_t = \mathrm{Attn}(\mathbf{h}_t,\, \{\mathbf{c}_s \mid s \in \mathcal{T}_t\})\,. \tag{3}$$

This reduces the dominant attention cost from $\mathcal{O}(L^2)$ to $\mathcal{O}(Lk)$. The interface between the two components is exactly the selected token set $\mathcal{T}_t$, which makes the indexer the natural target for further acceleration without touching Sparse MLA.

Indexer in HISA. HISA replaces the flat prefix scan with a two-stage coarse-to-fine search. The prefix is partitioned into $M = \lceil L/B \rceil$ contiguous blocks $\mathcal{B}_1, \ldots, \mathcal{B}_M$ of size B, and each block is summarized by a representative key obtained via mean pooling:

$$\tilde{\mathbf{k}}^I_b = \mathrm{Pool}\big(\{\mathbf{k}^I_s \mid s \in \mathcal{B}_b\}\big)\,. \tag{4}$$

In the first stage, the same indexing queries $\mathbf{q}^I_{t,j}$ and gating weights $w^I_{t,j}$ as in DSA are reused to score the pooled keys, and the top-m blocks are selected:

$$J_{t,b} = \sum_{j=1}^{H^I} w^I_{t,j} \cdot \mathrm{ReLU}\Big(\mathbf{q}^I_{t,j} \cdot \tilde{\mathbf{k}}^I_b\Big)\,, \quad \mathcal{C}_t = \mathrm{TopK}(J_{t,:},\, m). \tag{5}$$

Following (Lu et al., 2025), the first and last blocks are always included in $\mathcal{C}_t$ to retain the attention sink and local context. The candidate token set is then $\Omega_t = \bigcup_{b \in \mathcal{C}_t} \mathcal{B}_b$.

In the second stage, the original DSA scoring (Eq. 1) is applied within Ω_t, and the final top-k tokens

$$\mathcal{T}_t = \mathrm{TopK}(\{I_{t,s} \mid s \in \Omega_t\},\, k) \tag{6}$$

are passed to the unmodified Sparse MLA operator.

4 METHOD

As established in Section 3, the DSA indexer scores every prefix token with all H^I heads, giving a per-query cost of $\mathcal{O}(H^I L)$ that already dominates the indexer kernel even after FP8 quantisation, the ReLU non-linearity, and the small indexer dimension used by DeepSeek-V3.2 (DeepSeek-AI, 2025). The H^I heads cannot simply be collapsed into one: each specialises in a different relevance pattern, so reducing H^I measurably degrades retrieval. Our starting point is the observation that this expressiveness is needed only *in aggregate*—across all queries and across the prefix—whereas any single query is well served by a small, query-dependent subset of heads. The MISA method (Section 4.1) exploits this by routing such a subset on cheap block-level statistics; a hierarchical extension MISA† (Section 4.2) re-introduces the full head pool only on a small candidate set, improving agreement with DSA's selected token set.

4.1 MISA: MIXTURE OF INDEXER EXPERTS

We propose MISA (Mixture of Indexer Sparse Attention), shown alongside DSA in Figure 1. MISA treats the H^I indexer heads as a pool of experts and uses a lightweight router to select a small subset of $h \ll H^I$ active heads per query, in the spirit of MoE routing. The selected experts then perform the token-level scoring, while the routing decision itself is computed on cheap block-level statistics.

Block-pooled router. Following Eq. 4, the prefix is partitioned into $M = \lceil L/B \rceil$ blocks $\mathcal{B}_1, \ldots, \mathcal{B}_M$ and each block is summarized by a pooled indexing key $\tilde{\mathbf{k}}^I_b$. For query position t, the router computes per-head per-block affinities, weighted by the same gating coefficients $w^I_{t,j}$ used in the DSA score, and aggregates them across blocks to estimate the importance of each head:

$$A_{t,j,b} = w^I_{t,j} \cdot \mathrm{ReLU}\Big(\mathbf{q}^I_{t,j} \cdot \tilde{\mathbf{k}}^I_b\Big), \qquad E_{t,j} = \frac{1}{M}\sum_{b=1}^{M} |A_{t,j,b}|, \tag{7}$$

and selects the top-h heads as the active expert set:

$$\mathcal{H}_t = \mathrm{TopK}_j(E_{t,j},\, h)\,. \tag{8}$$

Including $w^I_{t,j}$ in $A_{t,j,b}$ makes $E_{t,j}$ a direct estimate of how much head j would contribute to the final aggregated score $I_{t,s}$, rather than an unweighted similarity. The router operates on the $M \ll L$ pooled keys with all H^I heads, so its cost is $\mathcal{O}(H^I M)$ per query. Crucially, since the router only needs to decide *which heads are relevant* for the current query rather than which regions to keep, MISA can use a much coarser block partition than HISA: in our experiments, B is set an order of magnitude larger than HISA's, which keeps M small and makes the routing overhead negligible compared to the subsequent token-level scoring.

It is worth contrasting the role of block pooling here with that in HISA. Both methods compute the same per-head per-block affinities $A_{t,j,b}$ from pooled keys, but they reduce them along orthogonal axes: HISA aggregates *across heads* to obtain a per-block score and selects the top-m blocks, whereas MISA aggregates *across blocks* to obtain a per-head importance and selects the top-h heads. In other words, block pooling is used here purely as a cheap proxy that avoids materializing the full $\mathcal{O}(H^I L)$ tensor of head–token products when estimating which heads matter for the current query.

Sparse token scoring with active experts. Given $\mathcal{H}_t$, only the active heads compute the token-level score:

$$\hat{I}_{t,s} = \sum_{j \in \mathcal{H}_t} w^I_{t,j} \cdot \mathrm{ReLU}\big(\mathbf{q}^I_{t,j} \cdot \mathbf{k}^I_s\big)\,, \tag{9}$$

and the final token set

$$\mathcal{T}_t = \mathrm{TopK}_s\Big(\hat{I}_{t,:},\, k\Big) \tag{10}$$

is passed to the unmodified Sparse MLA operator. The per-query indexer cost is reduced from the DSA baseline $\mathcal{O}(H^I L)$ to $\mathcal{O}(H^I M + hL)$, where $M = \lceil L/B \rceil \ll L$ and $h \ll H^I$. Unless otherwise specified, we use $h = 8$ on top of $H^I = 64$ (DeepSeek-V3.2), so the dominant hL term is reduced by $8\times$ or $4\times$ relative to DSA while the diversity of patterns available to the indexer is preserved—every head remains in the pool, and routing only chooses which ones to consult on each query.

Why MoE in the indexer. Standard MoE routes the FFN computation while keeping attention dense. MISA instead routes *which similarity patterns are used to select tokens*; the downstream attention itself remains dense over the chosen $\mathcal{T}_t$. The key empirical observation that makes this practical is that the relevant indexer heads of a query change slowly across the prefix, so a coarse block-level estimate is sufficient to predict them, and the heavy token-level scan only needs to consult a small subset of heads.

4.2 HIERARCHICAL MISA: COARSE-TO-FINE INDEXING

The single-stage MISA above already cuts indexer cost substantially. We further extend it to a coarse-to-fine variant, denoted MISA†, in which a cheap MoE pass first prunes the prefix to a candidate set, and a second pass refines the candidates with the full DSA indexer. Crucially, the second stage of MISA† uses the *DSA indexer* (Eq. 1) on the candidate tokens; we do not adopt HISA's block-level top-m filtering at any stage.

In the coarse stage, the MoE scoring of Eq. 9 is used to select an enlarged candidate set

$$\Omega_t = \mathrm{TopK}_s\Big(\hat{I}_{t,:},\, k'\Big)\,, \quad k' > k. \tag{11}$$

In the fine stage, the original DSA scoring is applied within Ω_t using all H^I heads, and the final tokens

$$\mathcal{T}_t = \mathrm{TopK}_s(\{I_{t,s} \mid s \in \Omega_t\}, k) \tag{12}$$

feed Sparse MLA.

Comparison with HISA. Both MISA† and HISA have a two-stage coarse-to-fine structure, but the granularity of the coarse pass is fundamentally different. HISA's coarse pass operates at the block level: pooling discards intra-block variation and a block must be kept or discarded as a whole, so the candidate set Ω_t is always a union of full blocks. MISA†, by contrast, keeps the coarse pass at full token granularity (Eq. 11) and reduces compute via head-level routing instead of block-level filtering, so tokens within the same block are still ranked individually. The fine pass is also stricter: it re-applies the original DSA scoring using all H^I heads on Ω_t, whereas HISA's fine pass operates on the union of selected blocks without revisiting the head set. Empirically, the final token set selected by MISA† exhibits higher IoU with DSA's top-k set than HISA in the comparison reported in Appendix C. This intrinsic agreement complements the downstream results in Sections 5.1 and 5.2.

5 Experimental results

We evaluate MISA as an inference-time replacement for the DSA indexer without additional training. Our main evaluation combines long-context quality on RULER, NIAH retrieval accuracy, and end-to-end prefill latency. Additional component profiling, concurrency experiments, and indexer-kernel microbenchmarks are provided in the appendix.

Models. RULER and NIAH evaluations use DeepSeek-V3.2 (DeepSeek-AI, 2025), which has $H^I = 64$ indexer heads.

Default hyperparameters. Unless otherwise specified, MISA uses router block size $B = 1024$ and $h = 8$ active heads to select $k = 2048$ tokens. The hierarchical variant, MISA†, first selects $k' = 8192$ candidates with the routed heads, then re-scores the candidates using all H^I DSA heads to retain the final $k = 2048$ tokens.

Baselines. DSA scores every prefix token using all H^I indexer heads. HISA (Xu et al., 2026) uses hierarchical block-to-token selection with block size $B = 128$. IndexCache (Bai et al., 2026) reuses selected token indices across layers. In our RULER evaluation, IndexCache$_{\frac{1}{2}}$ uses the uniform strategy to retain the indexer in one out of every two consecutive layers; the subscript denotes the fraction of layers that retain indexing. We also report IndexCache$_{\frac{1}{4}}$. The Block-Sparse baseline used in NIAH selects blocks of size $B = 128$ using pooled indexing keys, with the same final token budget $k = 2048$.

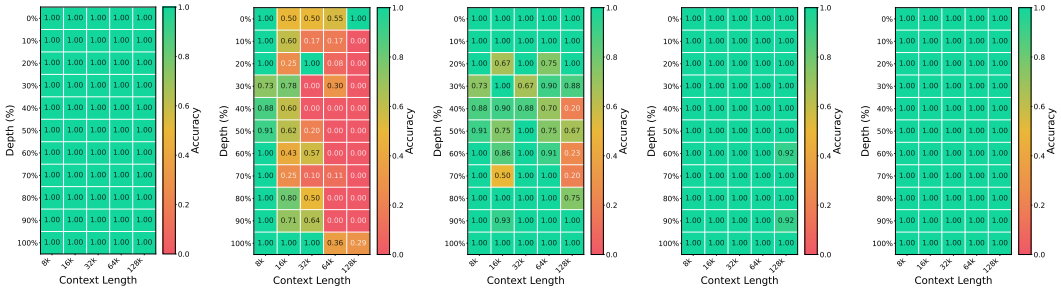
Evaluation. We evaluate DeepSeek-V3.2 on all 13 RULER subtasks (Hsieh et al., 2024), following the HELMET evaluation settings and prompts (Yen et al., 2025). We use 100 examples per subtask at each of six context lengths: 4K, 8K, 16K, 32K, 64K, and 128K. NIAH evaluates retrieval across context lengths up to 128K and needle depths from 0% to 100%. End-to-end efficiency is measured by mean time to first token (TTFT) in Section 5.3. Per-layer agreement with DSA's selected token set and additional routing ablations appear in Appendices C, D, E, and F.

5.1 RULER

Table 1 reports the mean score across the 13 RULER subtasks at each context length and the average across the six lengths. Across the six context lengths, MISA† achieves an average score of 94.94, second only to DSA at 95.62 among the evaluated methods and slightly above IndexCache$_{\frac{1}{2}}$ at 94.84. The single-stage MISA scores 93.73 on average. Both variants outperform HISA from 16K onward, with the largest gains at 128K: MISA and MISA† score 85.44 and 91.19, compared with 55.79 for HISA, corresponding to improvements of 29.65 and 35.40 points. The widening gap relative to HISA at longer contexts supports the effectiveness of head-axis routing, while the hierarchical refinement reduces the remaining quality gap to DSA.

Table 1: RULER scores for DeepSeek-V3.2. Each length averages all 13 subtasks with 100 examples per subtask. Avg. averages the six context lengths. Best and second-best scores in each column are bold and underlined, respectively.

Method	4K	8K	16K	32K	64K	128K	Avg.
DSA	**96.56**	96.19	96.17	**95.99**	**95.15**	**93.65**	**95.62**
IndexCache $\frac{1}{2}$	<u>96.49</u>	96.16	<u>96.21</u>	95.69	<u>94.71</u>	89.80	94.84
IndexCache $\frac{1}{4}$	96.41	**96.30**	**96.52**	94.13	94.30	89.65	94.55
HISA	96.41	96.21	94.97	83.96	67.92	55.79	82.54
MISA	96.15	96.15	96.14	95.52	92.95	85.44	93.73
MISA†	96.20	<u>96.25</u>	96.00	<u>95.74</u>	94.27	<u>91.19</u>	<u>94.94</u>



(a) DSA (original) (b) Block-Sparse (c) HISA (d) MISA (e) MISA†

Figure 2: Needle-in-a-Haystack retrieval accuracy on DeepSeek-V3.2 up to 128K context. The x-axis is context length and the y-axis is needle depth (0%–100%); greener is better. MISA and MISA† use $h = 8$ active heads in the routed scoring pass (vs. $H^I = 64$ for the baselines), with $k = 2048$ tokens selected; MISA† additionally refines a coarse candidate set of $k' = 8192$ using all H^I DSA heads.

5.2 Needle-in-a-Haystack retrieval accuracy

Figure 2 shows the Needle-in-a-Haystack (NIAH) retrieval accuracy of every method on DeepSeek-V3.2 at context lengths up to 128K. For each panel the x-axis sweeps the context length and the y-axis sweeps the needle depth from 0% (start of the haystack) to 100% (end). *DSA (original)* is the native dense indexer that scores every prefix token with all $H^I = 64$ heads (no token sparsification at the indexer level), and serves as the upper bound. Block-Sparse and HISA use $B = 128$ block partitioning with token budget $k = 2048$; MISA and MISA† both use $h = 8$ active heads in the routed scoring pass with $B = 1024$. MISA selects $k = 2048$ tokens in a single MoE-routed pass, while MISA† first selects a candidate set of size $k' = 8192$ with the same router and then re-ranks those candidates with all H^I DSA heads to keep the final $k = 2048$.

Both MISA variants reproduce the near-perfect green grid of DSA across the full depth–length plane, in stark contrast to Block-Sparse, whose block-uniform selection leaves visible accuracy holes at intermediate depths once the context exceeds $\sim$32K. HISA closes most of those holes but still shows minor degradations at the deepest needle positions, where its block-level top-m filtering can occasionally drop the block containing the needle. MISA†, in particular, is essentially indistinguishable from DSA in this evaluation. Its routed scan uses $h = 8$ of the $H^I = 64$ heads, while candidate refinement incurs additional computation using all H^I heads.

5.3 End-to-end prefill latency

We measure end-to-end prefill efficiency by mean time to first token (TTFT) over 10 measured samples following 10 separate warm-up samples. We evaluate a pruned 8-layer model at 256K and 512K and the full model at 256K; the full model runs out of memory at 512K. All numbers in this section are measured with the maximum request concurrency fixed to 4; Appendix A sweeps

Table 2: End-to-end mean TTFT in milliseconds (lower is better). **Full model** denotes the complete DeepSeek-V3.2 model; **8-layer** denotes its first 8 layers. Parentheses show speedup over DSA under the same configuration. The maximum request concurrency is fixed to 4 in all rows. The full model cannot run at 512K because of out-of-memory errors.

Model	Length	DSA	MISA†	MISA
8-layer	256K	24819.97	21186.45 (1.17×)	17435.64 (1.42×)
8-layer	512K	55470.99	37231.88 (1.49×)	35284.77 (1.57×)
Full model	256K	181464.46	142520.72 (1.27×)	127877.73 (1.42×)

this setting over $\{1, 2, 4, 8\}$. Speedup is the ratio of DSA to method mean TTFT under the same configuration.

Both variants reduce mean TTFT (Table 2). From 256K to 512K, 8-layer speedups increase from $1.17\times$ to $1.49\times$ for MISA† and from $1.42\times$ to $1.57\times$ for MISA. Full-model speedups at 256K are $1.27\times$ and $1.42\times$, respectively. Component timings follow below; concurrency results appear in Appendix A.

5.4 Component latency breakdown

Table 3 reports per-sample component totals: accumulated times over 10 measured samples divided by 10, in seconds. These are neither per-layer nor per-kernel timings and should not be summed to obtain TTFT. Indexer total includes routing, all Logits stages, and Top-k. Sparse MLA times are similar across methods, which share a final budget of 2048 tokens per query. The largest reduction is in the indexer: at 512K, DSA's 28.68s drops to 17.32s for MISA† and 15.60s for MISA (approximately $1.66\times$ and $1.84\times$). Routing costs 0.11s at 256K and 0.21s at 512K. The breakdown also shows a trade-off: both routed variants reduce Logits time but spend more time on Top-k than DSA. The two-stage variant additionally performs fine scoring on its coarse-stage candidates using all original indexer heads.

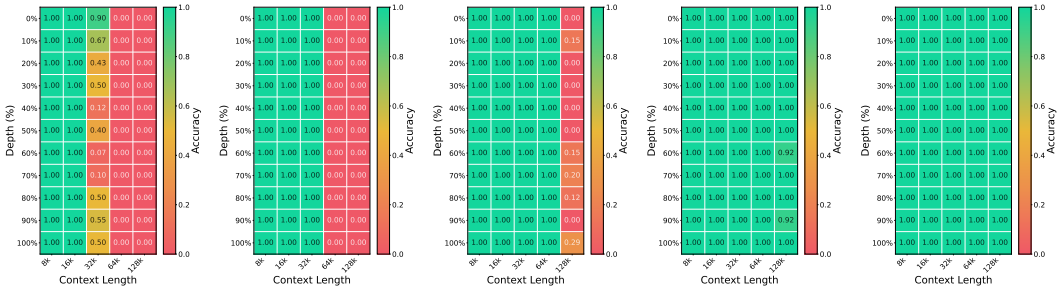
Table 3: Mean per-sample total component time in seconds, averaged over 10 measured samples and rounded to two decimal places. Indexer total includes Routing, all Logits stages, and Top-k; separately rounded components may differ from the total by 0.01s. For MISA†, Logits stage 1 and stage 2 denote coarse and fine scoring, respectively; DSA and MISA each use a single scoring pass, shown in stage 1. A dash denotes an unused component.

Length	Component	DSA	MISA†	MISA
256K	Routing	–	0.11	0.11
	Logits stage 1	8.31	3.53	3.51
	Logits stage 2	–	0.88	–
	Top-k	1.18	2.23	1.76
	Indexer total	9.49	6.75	5.38
	Sparse MLA	2.08	1.84	1.91
512K	Routing	–	0.21	0.21
	Logits stage 1	25.33	10.71	10.67
	Logits stage 2	–	0.95	–
	Top-k	3.35	5.45	4.72
	Indexer total	28.68	17.32	15.60
	Sparse MLA	2.28	2.02	2.07

5.5 Ablation: number of active heads

Figure 3 sweeps the number of active heads $h \in \{1, 2, 4, 8, 16\}$ on DeepSeek-V3.2 with $B = 1024$ and $k = 2048$ fixed, evaluated on NIAH at 128K. With $h = 1$ or $h = 2$, the heatmaps show visible accuracy holes, suggesting insufficient coverage of relevance patterns in DSA's $H^I = 64$-head pool.

Setting $h = 4$ mitigates most deficiencies but remains suboptimal, particularly at 128K. From $h = 8$ onwards, the heatmap becomes essentially indistinguishable from the dense 64-head indexer, with $h = 16$ providing no further gain despite using twice the compute of $h = 8$. We therefore use $h = 8$ as the default, since it is the smallest tested head budget that yields an essentially DSA-like heatmap in this 128K NIAH ablation. Additional ablations examine the coarse-stage active-head count in MISA† (Appendix D), the router scoring function (Appendix E), and the router block size (Appendix F).



(a) $h = 1$ (b) $h = 2$ (c) $h = 4$ (d) $h = 8$ (default) (e) $h = 16$

Figure 3: Ablation on the number of active heads h used by the MISA router on DeepSeek-V3.2 ($H^I = 64$, $B = 1024$, $k = 2048$). Each panel is a NIAH retrieval-accuracy heatmap at 128K context, with context length on the x-axis and needle depth (0%–100%) on the y-axis; greener is better.

6 Conclusion

We presented MISA, which serves as an MoE-style replacement for the DSA indexer. A lightweight block-pooled router selects a query-dependent subset of $h \ll H^I$ active heads, and only those heads run the heavy token-level scan. This single change cuts the dominant per-token cost of fine-grained sparse attention from $\mathcal{O}(H^I L)$ to $\mathcal{O}(hL + H^I M)$ while preserving the full diversity of the indexer pool, because every head remains available—routing simply chooses which ones to consult on each token. A coarse-to-fine extension, MISA†, additionally re-ranks an enlarged routed candidate set with all original DSA indexer heads, improving agreement with DSA's selected token set.

Our experiments characterize the quality–efficiency trade-off of routed indexing without additional training. On RULER, MISA† achieves an average score of 94.94 across 4K–128K, compared with 95.62 for DSA, and improves over HISA by 35.40 points at 128K. In end-to-end prefill evaluation, MISA achieves mean-TTFT speedups of $1.42\times$ on the full model at 256K and $1.57\times$ on a pruned 8-layer model at 512K. Component profiling and concurrency measurements further characterize the efficiency gains and their dependence on the evaluation setting.

7 Limitation

Our mean-TTFT evaluation measures prefill efficiency, not decoding latency or serving throughput. Full-model results cover only 256K: out-of-memory errors at 512K require a pruned 8-layer configuration, whose results do not establish full-model performance at that length. Reduced indexer computation does not eliminate KV-cache traffic: routed heads still scan the prefix, and downstream sparse attention retains the same token budget. All methods use pretrained models without finetuning; joint router–indexer training remains future work. Such training may improve the remaining quality–efficiency trade-off, but its benefits are not established by the present inference-only experiments.

AI USE STATEMENT

In this work, we used generative AI tools to assist with drafting portions of the manuscript, polishing its language for clarity and readability, and generating code. All AI-assisted manuscript content and generated code were personally reviewed and checked by the authors before inclusion or use, and revised or corrected as needed. The authors take full responsibility for the final content of this work, including its text, claims, experimental results, code, and other artifacts, regardless of whether generative AI tools assisted in their creation.

REFERENCES

Anthropic. Introducing Claude Opus 4.7. Technical report, Anthropic, 2026. URL https://www.anthropic.com/news/claude-opus-4-7. Context window: 1,000,000 tokens.

Yushi Bai, Qian Dong, Ting Jiang, Xin Lv, Zhengxiao Du, Aohan Zeng, Jie Tang, and Juanzi Li. IndexCache: Accelerating sparse attention via cross-layer index reuse. *arXiv preprint arXiv:2603.12201*, 2026. URL https://arxiv.org/abs/2603.12201.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020. URL https://arxiv.org/abs/2004.05150.

Zhuoming Chen, Ranajoy Sadhukhan, Zihao Ye, Yang Zhou, Jianyu Zhang, Niklas Nolte, Yuandong Tian, Matthijs Douze, Leon Bottou, Zhihao Jia, and Beidi Chen. MagicPIG: LSH sampling for efficient LLM generation. *arXiv preprint arXiv:2410.16179*, 2024. URL https://arxiv.org/abs/2410.16179.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019. URL https://arxiv.org/abs/1904.10509.

Damai Dai, Chengqi Deng, Chenggang Zhao, R.X. Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y. Wu, et al. DeepSeekMoE: Towards ultimate expert specialization in mixture-of-experts language models. *arXiv preprint arXiv:2401.06066*, 2024. URL https://arxiv.org/abs/2401.06066.

DeepSeek-AI. Deepseek-v3.2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*, 2025. URL https://arxiv.org/abs/2512.02556.

DeepSeek-AI. DeepSeek-V4: Towards highly efficient million-token context. *Technical Report, DeepSeek*, 2026. URL https://huggingface.co/deepseek-ai/DeepSeek-V4-Pro/resolve/main/DeepSeek_V4.pdf.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022. URL https://arxiv.org/abs/2101.03961.

Yizhao Gao, Zhichen Zeng, Dayou Du, Shijie Cao, Hayden Kwok-Hay So, Ting Cao, Fan Yang, and Mao Yang. SeerAttention: Learning intrinsic sparse attention in your LLMs. *arXiv preprint arXiv:2410.13276*, 2024. URL https://arxiv.org/abs/2410.13276.

Google DeepMind. Gemini 3: A new era of intelligence. Technical report, Google, 2026. URL https://blog.google/technology/google-deepmind/gemini-3/. Context window: 1,048,576 tokens.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. Ruler: What's the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*, 2024. URL https://arxiv.org/abs/2404.06654.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024. URL https://arxiv.org/abs/2401.04088.

Peng Jin, Bo Zhu, Li Yuan, and Shuicheng Yan. MoH: Multi-head attention as mixture-of-head attention. *arXiv preprint arXiv:2410.11842*, 2024. URL https://arxiv.org/abs/2410.11842.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. GShard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*, 2020. URL https://arxiv.org/abs/2006.16668.

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. *arXiv preprint arXiv:2501.08313*, 2025. URL https://arxiv.org/abs/2501.08313.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. SnapKV: LLM knows what you are looking for before generation. *arXiv preprint arXiv:2404.14469*, 2024. URL https://arxiv.org/abs/2404.14469.

Enzhe Lu, Zhejun Jiang, Jingyuan Liu, Yulun Du, Tao Jiang, Chao Hong, Shaowei Liu, Weiran He, Enming Yuan, Yuzhi Wang, Zhiqi Huang, Huan Yuan, Suting Xu, Xinran Xu, Guokun Lai, Yanru Chen, Huabin Zheng, Junjie Yan, Jianlin Su, Yuxin Wu, Neo Y. Zhang, Zhilin Yang, Xinyu Zhou, Mingxing Zhang, and Jiezhong Qiu. MoBA: Mixture of block attention for long-context LLMs. *arXiv preprint arXiv:2502.13189*, 2025. URL https://www.arxiv.org/abs/2502.13189.

Moonshot AI. Kimi K2: Open agentic intelligence. Technical report, Moonshot AI, 2025. URL https://github.com/MoonshotAI/Kimi-K2.

OpenAI. Introducing GPT-5.5. Technical report, OpenAI, 2026. URL https://openai.com/index/introducing-gpt-5-5/. API context window: 1,050,000 tokens.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017. URL https://arxiv.org/abs/1701.06538.

Jiaming Tang, Yilong Zhao, Kan Zhu, Guangxuan Xiao, Baris Kasikci, and Song Han. Quest: Query-aware sparsity for efficient long-context LLM inference. In *International Conference on Machine Learning*, 2024. URL https://arxiv.org/abs/2406.10774.

Chaojun Xiao, Pengle Zhang, Xu Han, Guangxuan Xiao, Yankai Lin, Zhengyan Zhang, Zhiyuan Liu, Song Han, and Maosong Sun. InfLLM: Training-free long-context extrapolation for LLMs with an efficient context memory. *arXiv preprint arXiv:2402.04617*, 2024a. URL https://arxiv.org/abs/2402.04617.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*, 2024b. URL https://arxiv.org/abs/2309.17453.

Yufei Xu, Fanxu Meng, Fan Jiang, Yuxuan Wang, Ruijie Zhou, Zhaohui Wang, Jiexi Wu, Zhixin Pan, Xiaojuan Tang, Wenjie Pei, Tongxuan Liu, Di Yin, Xing Sun, and Muhan Zhang. HISA: Efficient hierarchical indexing for fine-grained sparse attention. *arXiv preprint arXiv:2603.28458*, 2026. URL https://www.arxiv.org/abs/2603.28458v3.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Howard Yen, Tianyu Gao, Minmin Hou, Ke Ding, Daniel Fleischer, Peter Izsak, Moshe Wasserblat, and Danqi Chen. Helmet: How to evaluate long-context models effectively and thoroughly. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.02694.

Jingyang Yuan, Huazuo Gao, Damai Dai, Junyu Luo, Liang Zhao, Zhengyan Zhang, Zhenda Xie, Y.X. Wei, Lean Wang, Zhiping Xiao, Yuqing Wang, Chong Ruan, Ming Zhang, Wenfeng Liang, and Wangding Zeng. Native sparse attention: Hardware-aligned and natively trainable sparse attention. *arXiv preprint arXiv:2502.11089*, 2025. URL https://arxiv.org/abs/2502.11089.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems*, 2020. URL https://arxiv.org/abs/2007.14062.

Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chengxing Xie, Chenzheng Zhu, et al. GLM-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026. URL https://arxiv.org/abs/2602.15763.

Xiaofeng Zhang, Yikang Shen, Zeyu Huang, Jie Zhou, Wenge Rong, and Zhang Xiong. Mixture of attention heads: Selecting attention heads per token. In *Conference on Empirical Methods in Natural Language Processing*, 2022. URL https://arxiv.org/abs/2210.05144.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. H_2O: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2306.14048.

A END-TO-END PERFORMANCE ACROSS CONCURRENCY LEVELS

We evaluate end-to-end latency using the same 8-layer slice of DeepSeek-V3.2 (its first 8 layers) as in Section 5.3, at a fixed input length of 512K while varying the maximum request concurrency over $\{1, 2, 4, 8\}$. This parameter controls the maximum number of concurrent requests and should not be interpreted as a fixed execution batch size. We report the measured latency values and the corresponding speedup over DSA at each concurrency level.

Table 4: Reported end-to-end latency (ms) at 512K under different maximum request concurrency levels. Parentheses indicate speedup over DSA at the same concurrency. The concurrency-4 row reproduces the 8-layer 512K row of Table 2.

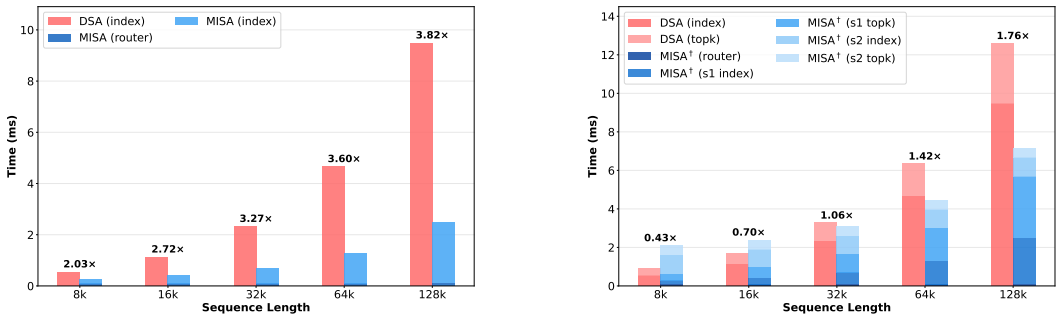
Max concurrency	DSA	MISA†	MISA
1	21351.64	13970.86 ($1.53\times$)	13230.90 ($1.61\times$)
2	29439.20	19542.97 ($1.51\times$)	18488.43 ($1.59\times$)
4	55470.99	37231.88 ($1.49\times$)	35284.77 ($1.57\times$)
8	103024.19	79358.59 ($1.30\times$)	71121.32 ($1.45\times$)

Both variants remain faster than DSA at every tested concurrency level (Table 4). The speedups range from $1.30\times$ to $1.53\times$ for MISA† and from $1.45\times$ to $1.61\times$ for MISA. The arithmetic means of the four per-concurrency speedup ratios, computed before rounding, are $1.46\times$ and $1.56\times$, respectively. Although the benefit decreases at the highest concurrency, both methods retain a reduction in reported latency.

B INDEXER KERNEL MICROBENCHMARK

Figure 4 compares the wall-clock latency of the indexer kernel for DSA and MISA, under both the 1-stage (Fig. 4a) and 2-stage (Fig. 4b, marked as MISA†) configurations. DSA scores every prefix token using all $H^I = 64$ indexer heads, resulting in a computational cost that grows as $\mathcal{O}(LH^I)$. In contrast, MISA activates only $h = 8$ heads per query, scoring the prefix with complexity $\mathcal{O}(Lh + MH^I)$, where the second term corresponds to the (negligible) router overhead applied on $M = \lceil L/B \rceil \ll L$ pooled keys. In the 2-stage setting, the complexity becomes $\mathcal{O}(Lh + MH^I + k'H^I)$, where k' is fixed to 8192 and accounts for the additional scoring cost in the second stage.

In the 1-stage setting, the MISA kernel is consistently faster than DSA across the entire sequence-length range. In the 2-stage setting, it also outperforms DSA when the sequence length exceeds 32k. The asymptotic $H^I/h = 8\times$ reduction in head–token products is not fully reflected in wall-clock latency—factors such as memory traffic, load imbalance introduced by the routed expert set, and a small but non-zero router overhead all reduce the theoretical speedup. Nevertheless, our TileLang implementation achieves approximately a $3.82\times$ speedup over DSA's original indexer kernel for long contexts in this isolated microbenchmark. This is an indexer-kernel speedup, not a full-model TTFT speedup; the latter is evaluated separately in Section 5.3.



(a) MISA

(b) MISA†

Figure 4: Indexer-kernel latency for DSA and MISA, as a function of prefix length. MISA uses $h = 8$ active heads with router block size $B = 1024$ and selects $k = 2048$ tokens. **(a)** MISA latency. **(b)** MISA† latency (first stage selects $k' = 8192$ tokens). Lower is better.

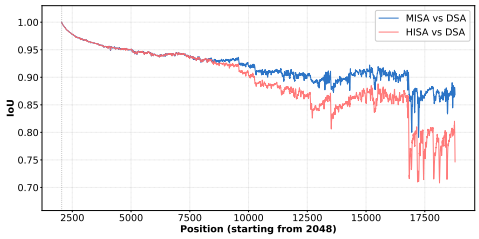
C PER-LAYER AGREEMENT WITH THE DSA TOP-k

As an intrinsic check that head-level routing does not distort the selected token set, we measure the per-layer Intersection-over-Union (IoU) between each method's selected tokens and the DSA top-2048 *reference* set. Concretely, on the long subset of LSHT (10 examples, all T layers, on DeepSeek-V3.2), we run DSA with all $H^I = 64$ indexer heads to obtain the reference token set $\mathcal{T}_t^{\mathrm{DSA}}$, and for every other method we report

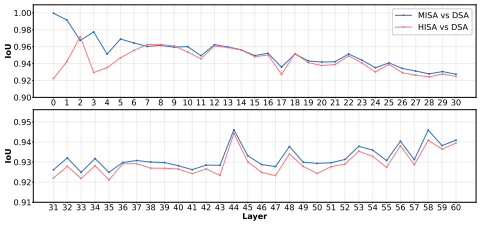
$$\mathrm{IoU}_t^{(\ell)} = \frac{\left|\mathcal{T}_t \cap \mathcal{T}_t^{\mathrm{DSA}}\right|}{\left|\mathcal{T}_t \cup \mathcal{T}_t^{\mathrm{DSA}}\right|}, \qquad \ell = 1, \ldots, T. \tag{13}$$

The IoU becomes meaningful only once the prefix length meets or exceeds the token budget; therefore, we initiate the curves at position $t = 2048$. Both MISA and HISA use the identical final budget of $k = 2048$. We evaluate MISA in a two-stage configuration, denoted as MISA†: the shared router first generates a candidate set of size $k' = 8192$, which is subsequently re-scored by all $H^I = 64$ DSA heads to select the final $k = 2048$ tokens.

Figure 5 summarizes the results. In each panel, the Intersection-over-Union (IoU) between the token sets selected by MISA† and the DSA baseline is represented by the blue curve, whereas the red curve corresponds to the IoU between HISA and the DSA baseline. Figure 5a shows how the average IoU varies with sequence length, and Figure 5b compares the IoU layer by layer. MISA† consistently outperforms HISA across sequence lengths and at every layer. This agreement with DSA's selected token set is consistent with the downstream quality observed on RULER and NIAH (Sections 5.1 and 5.2).



(a) IoU vs. token position

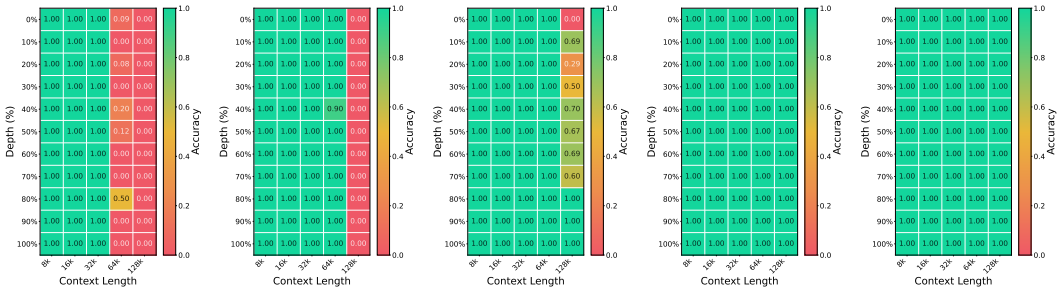


(b) IoU vs. layer index

Figure 5: Per-layer Intersection-over-Union between the indexer-selected token set and the DSA top-2048 golden set on LSHT. Curves are plotted for HISA and MISA-stage-2 (the fine pass of MISA†, which re-ranks the $k' = 8192$ candidates with all H^I heads to keep $k = 2048$). **(a)** IoU as a function of token position, averaged across T layers; the position axis starts at 2048 since IoU is only well-defined once the prefix has at least k tokens. **(b)** IoU as a function of layer index.

D ABLATION: ACTIVE HEADS' NUMBER OF MISA†

Figure 6 repeats the same sweep for the hierarchical variant MISA†, where the routed pass with h heads now selects an enlarged candidate set of size $k' = 4k = 8192$, and the full $H^I = 64$-head DSA indexer then re-ranks this candidate set to extract the final $k = 2048$ tokens. Because the routed stage only has to keep the relevant tokens *inside* the candidate set rather than pinpoint the exact top-k, the DSA refinement can compensate for very aggressive head reduction: A key observation is that the hierarchical variant MISA† with $h = 1$ achieves comparable quality to the single-stage MISA with $h = 2$. Similarly, MISA† with $h = 2$ matches the performance of the single-stage variant with $h = 4$. Even $h = 4$ delivers strong results at the full 128K context length. The hierarchical pipeline is therefore highly tolerant to under-routing in stage one, which is exactly the regime in which MISA† achieves the strongest indexer fidelity (cf. Figure 5) while still using a small h in the heavy token-level scan.

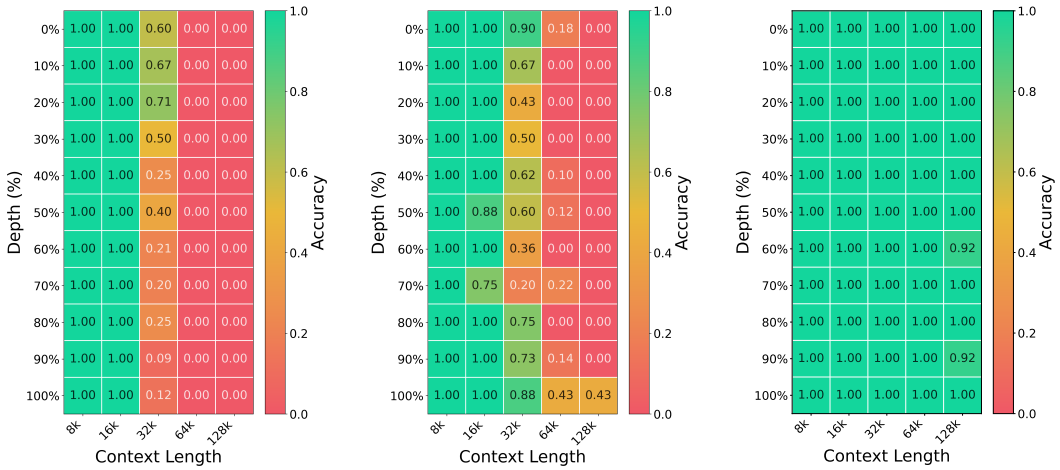


(a) $h = 1$ (b) $h = 2$ (c) $h = 4$ (d) $h = 8$ (default) (e) $h = 16$

Figure 6: Ablation on the number of active heads h used by the *coarse* routed stage of the hierarchical variant MISA† on DeepSeek-V3.2 ($H^I = 64$, $B = 1024$). The routed stage selects an enlarged candidate set of $k' = 4k = 8192$ tokens, which is then re-ranked by the full $H^I = 64$-head DSA indexer to obtain the final $k = 2048$ tokens. Each panel is a NIAH retrieval-accuracy heatmap at 128K context, with context length on the x-axis and needle depth on the y-axis. In comparison to the single-stage MISA results presented in Figure 3, the hierarchical pipeline demonstrates markedly greater robustness to a reduced number of routing heads (h). A key finding is that configurations with $h = 1$ and $h = 2$ in this two-stage framework achieve performance parity with the single-stage MISA utilizing twice as many heads (i.e., $h = 2$ and $h = 4$, respectively), because the second stage can recover any tokens that the routed stage missed within the enlarged candidate set.

E ABLATION: ROUTING SCORE

The router score $E_{t,j}$ in Eq. 7 is the only learning-free signal that decides which h heads are active for query t, so its choice is a critical design knob. We compare three candidates on DeepSeek-V3.2 with the default MISA setting ($h = 8$, $B = 1024$, $k = 2048$) and report NIAH accuracy at 128K (Figure 7). The first two are content-light proxies that can be computed without ever consulting the prefix: (a) using the gating weight $w^I_{t,j}$ alone selects the heads that the DSA aggregator already up-weights; (b) the ℓ_2 norm $\|\mathbf{q}^I_{t,j}\|_2$ picks the heads whose query directions are most "confident"; None of these signals depends on what is actually *in* the prefix that needs to be retrieved, and both two collapse on the harder regions of the NIAH grid. Variant (c), the proposed block-pooled attention $E_{t,j} = \frac{1}{M}\sum_b |w^I_{t,j}\,\mathrm{ReLU}(\mathbf{q}^I_{t,j}\cdot\tilde{\mathbf{k}}^I_b)|$, is the only score that aggregates query-to-prefix evidence into the routing decision, and is the only variant that fully recovers DSA's accuracy. This isolates the importance of *where* the routing signal comes from rather than just *which* heads are routed.



(a) $w^I_{t,j}$ (b) $\|\mathbf{q}^I_{t,j}\|_2$ (c) $\frac{1}{M}\sum_b |w^I_{t,j}\,\mathrm{ReLU}(\mathbf{q}^I_{t,j}\cdot\tilde{\mathbf{k}}^I_b)|$

Figure 7: Ablation on the head-importance score $E_{t,j}$ used by the MISA router on DeepSeek-V3.2 (Needle-in-a-Haystack accuracy at 128K). (a) the indexer gating weight alone; (b) the ℓ_2 norm of the query head; (c) the proposed block-attention score, averaged over the M pooled blocks of the prefix — the only variant that actually consults past content. The x-axis denotes context length and the y-axis the needle depth (0%–100%); greener is better.

F ABLATION: ROUTER BLOCK SIZE

Finally, we sweep the router block size B on DeepSeek-V3.2 with $h = 8$ and $k = 2048$ fixed, ranging B from 128 to the full prefix length $L = 131{,}072$ (a single global pooled key)—eleven values, spanning three orders of magnitude. Recall that the router cost is $\mathcal{O}(H^I M)$ with $M = \lceil L/B \rceil$ pooled keys, so B trades router cost against the spatial resolution of the head-importance estimate. Figure 8 shows the resulting NIAH heatmaps. Across this entire sweep, retrieval accuracy is largely insensitive to B: the heatmaps are visually indistinguishable from the dense DSA reference for the small to moderate block sizes, and only the very largest B values (where the router is forced to summarise the prefix into a handful of pooled keys) begin to lose enough locality to introduce a mild degradation at the deepest needle depths. The default $B = 1024$ sits comfortably inside the stable region while keeping $M = 128$ pooled keys at 128K context, which makes the router cost (a single $H^I \times M$ matmul per query) negligible compared to the subsequent token-level scoring.



Figure 8: Ablation on the router block size B on DeepSeek-V3.2 ($h = 8$, $k = 2048$). Each panel is a NIAH retrieval-accuracy heatmap at 128K context (context length on the x-axis, needle depth on the y-axis; greener is better). Accuracy is largely insensitive to B across the full sweep, and only degrades mildly at the largest block sizes where the router is forced into a near-global pool.