DOSA: DON'T SCORE ALL CERTIFIED SPARSE ATTENTION INDEXING

Anonymous authors
Paper under double-blind review

ABSTRACT

Sparse attention visits few keys, yet finding them can still require scoring every key with every indexer head. We introduce DOSA (*Don't Score All*), a lossless method for making this computation conditional while preserving exact token selection. Its central observation is that Top-K selection requires sufficient evidence to reject a candidate, not necessarily its full score. A seed set supplies a lower bound on the winning threshold. Nonnegative head projections and joint residual bounds upper-bound positive contributions, while activation-informed aggregation bounds the magnitude subtracted by negative heads. The certificate directs full scoring toward candidates that remain eligible; exact-score filtering further reduces writeback and selection work. We establish the certificate and its exact-selection conditions, and analyze how geometry and selection margins govern its effectiveness. Our GPU implementation accelerates the complete Indexer by up to $2.74\times$ over SGLang and $1.69\times$ over LiteTopK across English, Chinese, and code workloads. End-to-end evaluation on GLM-5.2 and DeepSeek-V3.2 achieves up to $1.94\times$ and $1.50\times$ time-to-first-token speedups over the two baselines, respectively.

1 INTRODUCTION

Conditional computation can scale model capacity while activating only part of the model for each input. Mixture-of-experts models apply this principle in feed-forward networks (Fedus et al., 2022); attention offers a complementary opportunity as contexts grow. Sparse attention visits few keys, yet identifying them can still require a dense scan. DeepSeek Sparse Attention (DSA) selects tokens with a multihead indexer followed by Top-K (DeepSeek-AI, 2025); each score requires dot products, ReLU activations, and a signed weighted reduction. This leaves a dense scoring bottleneck inside sparse attention, limiting the MoE-style separation of capacity and active computation. We take a step toward scalable attention by making token selection conditional while preserving the original routing rule.

Figure 1 illustrates the resulting bottleneck. At approximately one million tokens, the indexer accounts for 73.0% of prefill GPU time in GLM-5.2 and 90.8% in DeepSeek-V3.2, while its shares of logical matrix FLOPs are 47.4% and 83.2%, respectively. As context grows, choosing where to attend can dominate the cost of sparse attention itself. The gap between time and arithmetic shares also motivates reducing selection and memory traffic alongside score computation.

Recent systems accelerate selection: Guess-Verify-Refine (GVR) exploits temporal correlation in decoding (Cheng et al., 2026); DeepSelect streams materialized scores through a bounded buffer (Qian et al., 2026); and LiteTopK fuses scoring with candidate filtering to avoid dense score writeback (Yin et al., 2026). These paths still require all-head evaluation for every key. MISA instead screens with fewer heads and reranks candidates with the full indexer (Zhou et al., 2026), but reranking cannot recover excluded winners. Can we *avoid full-score computation while certifying that screening discards no winner?*

The principle is simple: *a token cannot enter Top-K if its score upper bound falls below a lower bound on the true Top-K threshold.* A subset of at least K visible keys supplies the latter through its Kth largest score, following established exact-retrieval principles (Turtle & Flood, 1995; Broder et al., 2003). The challenge is to construct a tight bound at substantially lower cost than the score itself. For the signed ReLU indexer, dropping positive-weight heads is unsafe; dropping negative-weight

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.

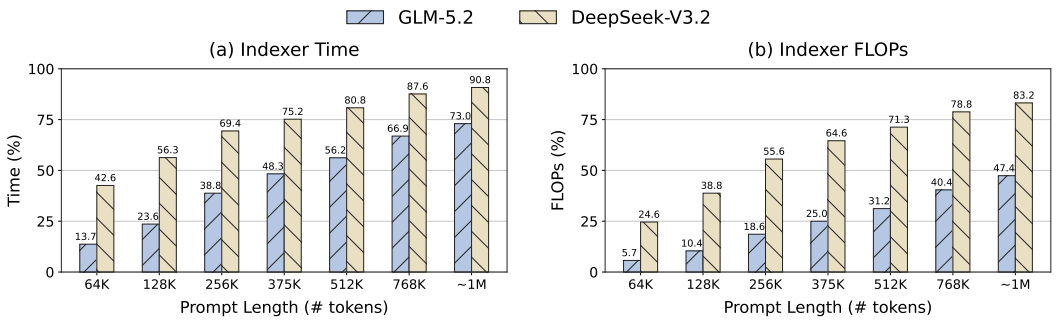


Figure 1: The indexer becomes a dominant cost of long-context prefill. Left: its fraction of prefill GPU time, including query/key preparation, scoring, and Top-K. Right: its fraction of theoretical matrix FLOPs. Each point averages two requests under SGLang; profiling and FLOP conventions are given in Appendix D.1.

heads is safe but can leave loose bounds. A useful certificate must exploit the geometry and activation structure of the heads.

DOSA, a lossless optimization, constructs token-specific bounds by evaluating selected heads and bounding the rest. Nonnegative projections reuse retained dot products, and joint residual bounds exploit dependence among omitted positive heads. An activation-informed aggregate lower-bounds the magnitude subtracted by omitted negative heads. These query-side quantities are reused across keys. Figure 2 shows how the certificate leads to three complementary savings: fewer score computations, fewer score writes, and less Top-K work. Survivors receive full original scores, and only scores meeting the seed-derived threshold are written for final selection.

Our contributions are a certificate tailored to the signed DSA score, an analysis connecting its slack to selection margins and head budgets, and a GPU realization that turns certified rejection into reduced computation and traffic. Across English, Chinese, and code workloads, DOSA outperforms both baselines in all 60 microbenchmark configurations, with up to $2.74\times$ complete-Indexer speedup over SGLang and $1.69\times$ over LiteTopK. At approximately 1M tokens, it achieves $1.86\times$ and $1.94\times$ time-to-first-token (TTFT) speedups over SGLang on GLM-5.2 and DeepSeek-V3.2, respectively, nearly halving end-to-end latency.

2 METHOD

2.1 PROBLEM FORMULATION

For a fixed query with H indexer heads, let $\mathbf{q}_h \in \mathbb{R}^D$ be the hth query vector, $w_h \in \mathbb{R}$ its weight, and $\mathbf{k}_i \in \mathbb{R}^D$ the key for token i. Write $[x]_+ = \max(x, 0)$. The indexer score is

$$s_i = \sum_{h=1}^{H} w_h \left[\mathbf{q}_h^\top \mathbf{k}_i\right]_+, \qquad i \in \Omega, \tag{1}$$

where Ω contains only visible keys.

For $|\Omega| \geq K$, our goal is to recover the K highest-scoring visible keys while avoiding full-head scoring for keys that cannot enter this set.

2.2 SEED THRESHOLD AND RETAINED HEADS

Choose distinct visible keys $\mathcal{S} \subseteq \Omega$, with $|\mathcal{S}| \geq K$, and set $\tau = \tau_K(\mathcal{S})$ to their Kth largest full score. The subset supplies K witnesses:

$$\tau_K(\mathcal{S}) \leq \tau_K(\Omega), \qquad s_i \leq U_i < \tau_K(\mathcal{S}) \implies i \notin \mathrm{TopK}(\Omega). \tag{2}$$

Thus $\tau_K(\mathcal{S})$ is a lower bound on the true Top-K threshold, and U_i is an upper bound on the token score: their strict separation certifies rejection. A poor seed weakens filtering, and additional exact scores can only raise the threshold lower bound.

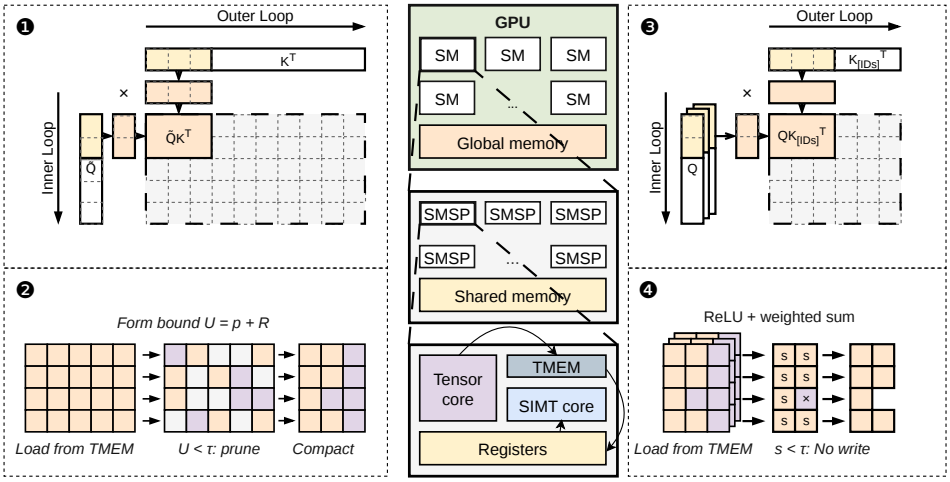


Figure 2: DOSA execution overview, read in order ❶–❹: ❶ screening dot products; ❷ bound evaluation, pruning, and candidate compaction; ❸ full-head dot products for survivors; ❹ original-score reduction and sparse writeback. Tensor Cores execute stages ❶ and ❸; SIMT threads execute stages ❷ and ❹ using accumulators from Tensor Memory (TMEM), with gray and purple data tiles indicating avoided computation and writeback, respectively. The center sketches GPU resources; seed initialization and final Top-K selection are omitted.

Bounding every head can leave all candidates above the threshold. We therefore directly evaluate selected heads whose contributions are difficult to bound tightly.

Let R be a set of $1 \le r < H$ retained heads. Define $\mathcal{H}_+ = \{h : w_h > 0\}$ and $\mathcal{H}_- = \{h : w_h < 0\}$. If $|\mathcal{H}_+| \le r$, retain all positive heads and fill remaining positions with negative heads prioritized by $|w_h| \, \|\mathbf{q}_h\|_2 \min(p_h, 1 - p_h)$, where p_h is their activation frequency on sampled seed keys. This favors strong heads whose activation is difficult to predict. Otherwise, retain the r strongest positive heads by $w_h \, \|\mathbf{q}_h\|_2$. Zero-weight heads need no contribution. Write $P = \mathcal{H}_+ \setminus R$ and $N = \mathcal{H}_- \setminus R$. The retained contribution

$$S_R(\mathbf{k}_i) = \sum_{j \in R} w_j \left[\mathbf{q}_j^\top \mathbf{k}_i\right]_+ \tag{3}$$

is evaluated directly. The omitted heads in P and N are not evaluated individually during screening; their contributions are bounded in the next two subsections. The proof holds for any R; its choice affects efficiency and tightness.

2.3 UPPER-BOUNDING OMITTED POSITIVE HEADS

For each $h \in P$, choose a retained anchor and a nonnegative projection:

$$j_h^\star = \arg\max_{j \in R} \frac{\left[\mathbf{q}_h^\top \mathbf{q}_j\right]_+}{\|\mathbf{q}_j\|_2}, \quad \beta_h = \frac{\left[\mathbf{q}_h^\top \mathbf{q}_{j_h^\star}\right]_+}{\|\mathbf{q}_{j_h^\star}\|_2^2}, \quad \mathbf{d}_h = \mathbf{q}_h - \beta_h \mathbf{q}_{j_h^\star}. \tag{4}$$

A zero-norm anchor has matching score and coefficient zero. ReLU subadditivity, positive homogeneity, and Cauchy–Schwarz give

$$\left[\mathbf{q}_h^\top \mathbf{k}_i\right]_+ \le \beta_h \left[\mathbf{q}_{j_h^\star}^\top \mathbf{k}_i\right]_+ + \|\mathbf{d}_h\|_2 \, \|\mathbf{k}_i\|_2 . \tag{5}$$

The projected part reuses an already computed dot product. The nonnegative projection minimizes the residual norm, yielding the smallest uniform norm allowance among single-anchor bounds of this form (Appendix B.1). Define

$$\widetilde{w}_j = w_j + \sum_{\substack{h \in P \\ j_h^\star = j}} w_h \beta_h, \qquad E_P = \sum_{h \in P} w_h \, \|\mathbf{d}_h\|_2 . \tag{6}$$

Then retained and omitted positive terms together are bounded by $\sum_{j\in R} \widetilde{w}_j[\mathbf{q}_j^\top \mathbf{k}_i]_+ + E_P \|\mathbf{k}_i\|_2$. The folded weights belong only to the screening plan; they do not modify model parameters. If P is empty, $E_P = 0$ and $\widetilde{w}_j = w_j$.

The omitted residuals also share the same key. Let $\mathbf{u}_h = w_h \mathbf{d}_h$ for $h \in P$. A joint allowance is

$$E_P^{\text{joint}} = \tfrac{1}{2}\left(\left\|\sum_{h\in P} \mathbf{u}_h\right\|_2 + \sqrt{\sum_{h,\ell\in P} |\mathbf{u}_h^\top \mathbf{u}_\ell|}\right) \le E_P. \tag{7}$$

It bounds $\sum_{h\in P}[\mathbf{u}_h^\top \mathbf{k}_i]_+$ by $E_P^{\text{joint}} \|\mathbf{k}_i\|_2$ and can therefore replace E_P. The improvement exploits residual geometry without additional key reads. Appendix A.1 proves this statement and its grouped form.

2.4 Lower-bounding omitted negative contributions

Negative heads subtract from the score. To construct an upper bound, we aggregate a subset $A \subseteq N$ and ignore the remaining heads; omitting negative contributions can only raise the bound. Define

$$\mathbf{a}_A = \sum_{h\in A} |w_h| \mathbf{q}_h, \qquad L_N(\mathbf{k}_i) = \left[\mathbf{a}_A^\top \mathbf{k}_i\right]_+ \le \sum_{h\in N} |w_h| \left[\mathbf{q}_h^\top \mathbf{k}_i\right]_+ . \tag{8}$$

ReLU subadditivity ensures that L_N does not exceed the true amount subtracted by the omitted negative heads. Why not simply include every head in A? An inactive head contributes zero after its own ReLU, but its negative dot product can cancel active heads' positive dot products before the aggregate ReLU. Including such heads can therefore weaken the negative-contribution estimate and loosen the score upper bound.

We use seed activation patterns to reduce this cancellation. For sampled seed keys, record $M_i = \{h \in N : \mathbf{q}_h^\top \mathbf{k}_i > 0\}$ and let p_h be the frequency of head h. We choose componentwise majority support, $A = \{h \in N : p_h \ge 1/2\}$. This minimizes a separable upper bound on normalized negative slack; jointly, the retained-negative priority above removes the largest remaining terms of that surrogate (Appendix B.2). If a new key has active set exactly A, the aggregate is exact. Other patterns remain safe and may give a looser bound. The frequencies guide efficiency, not correctness.

2.5 Score bounds and candidate refinement

Combining Equations (5)–(8) gives our score certificate:

$$\boxed{U_i = \sum_{j\in R} \widetilde{w}_j \left[\mathbf{q}_j^\top \mathbf{k}_i\right]_+ + E_P \|\mathbf{k}_i\|_2 - \left[\mathbf{a}_A^\top \mathbf{k}_i\right]_+ , \qquad s_i \le U_i.} \tag{9}$$

One aggregate direction replaces arbitrarily many omitted negative heads. Its utility depends on activation agreement, while the positive allowance depends on directional similarity.

Figure 2 maps the certificate to four stages. Stage ❶ reuses screening rows $\widetilde{Q}$ (retained and auxiliary) across key tiles; outer and inner loops traverse keys and query rows, respectively. Stage ❷ forms $U = p + R$: p combines the retained and aggregate terms in Equation (9), and the scalar $R = E_P \|\mathbf{k}_i\|_2$ is the residual allowance, distinct from the head set R. FP8 adds numerical guards (Appendix C).

Stage ❷ discards keys with $U_i < \tau$ and compacts survivor IDs. Stage ❸ gathers their keys and evaluates all H original heads. Stage ❹ applies the original ReLU and weights from Equation (1), retaining only scores $s_i \ge \tau$ for selection with the seed records. In the data tiles of stages ❷ and ❹, gray marks avoided computation and purple marks avoided writeback. Equality survives both filters; bounds never replace final scores.

With valid seed thresholds and no loss of live candidates, this procedure preserves exact Top-K selection; Appendix A gives the complete algorithm, formal guarantee (Theorem 1), and proof.

3 GPU Kernel Design

Certified pruning reduces arithmetic, but irregular survivor sets can limit GPU utilization and introduce scattered key accesses. During prefill, we organize DoSA into a regular screening pass followed by

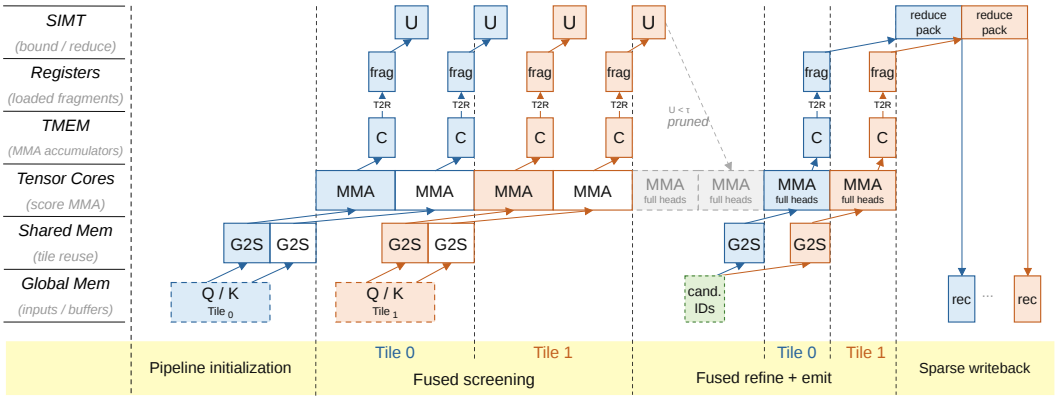


Figure 3: Data flow through screening, refinement, and sparse writeback. Blue and orange illustrate successive tiles within each stage, where key loading overlaps computation. SIMT threads evaluate bounds U from TMEM accumulators; gray full-head MMA blocks indicate work avoided through certified pruning. Compacted candidate IDs feed refinement, whose outputs are filtered before writeback. G2S and T2R denote global-to-shared and TMEM-to-register transfers.

sparse full-head refinement (Figure 3). Three design choices translate the certificate into efficient execution: grouping queries for Tensor Core computation, sharing key loads during refinement, and reducing the records passed to selection.

Query grouping and pipelined screening. We group queries with similar retained-head budgets into fixed-width tiles, enabling a small set of screening kernels to use dense Tensor Core MMA. Retained query rows, folded weights, and two FP8 auxiliary rows are prepared once and reused across key tiles. Exact scores of visible seed keys initialize each query's threshold, which remains fixed during the scan. Dedicated warps issue asynchronous key loads and MMA operations, while consumer warps evaluate bounds from TMEM accumulators. Buffered key tiles overlap loading with computation. Keys are rejected only when $U_i^{\mathrm{num}} < \tau$, and warp ballots pack the survivor decisions into compact masks for refinement.

Shared key access in refinement. Independent per-query gathers would repeatedly load keys selected by several queries. We therefore compact the survivor masks into shared and per-query refinement streams. Within a group of four queries, a key that survives at least two bounds and lies in the group's common visible range enters the shared stream; other candidates use per-query streams. A shared key load serves full-head scoring for the group, followed by the original ReLU, weights, and reduction. This organization trades some extra query–key evaluations for fewer key transfers. Each score is filtered against its query's threshold, preserving exact selection.

Compact writeback and exact selection. Refinement fuses full-score computation with threshold filtering: only scores $s_i^{\mathrm{ref}} \geq \tau$ produce output records. These records store compact monotone score codes with exact key IDs, reducing HBM traffic and selection input. After refinement, Top-K selection retains records strictly above the boundary code and resolves candidates at the boundary using their original scores. Together, screening reduces full-head arithmetic, shared refinement reuses key data, and compact records reduce the cost of writing and selecting scores.

Decoding. With few queries per step, decoding offers limited key reuse across query groups. We retain full FP8 scoring and exploit temporal coherence to reduce selection work. Distinct keys selected in recent steps provide witnesses whose current-query scores yield a conservative lower bound on the Top-K threshold. Scores at or above this bound are compacted for exact selection, which also maps the selected indices and updates the witnesses.

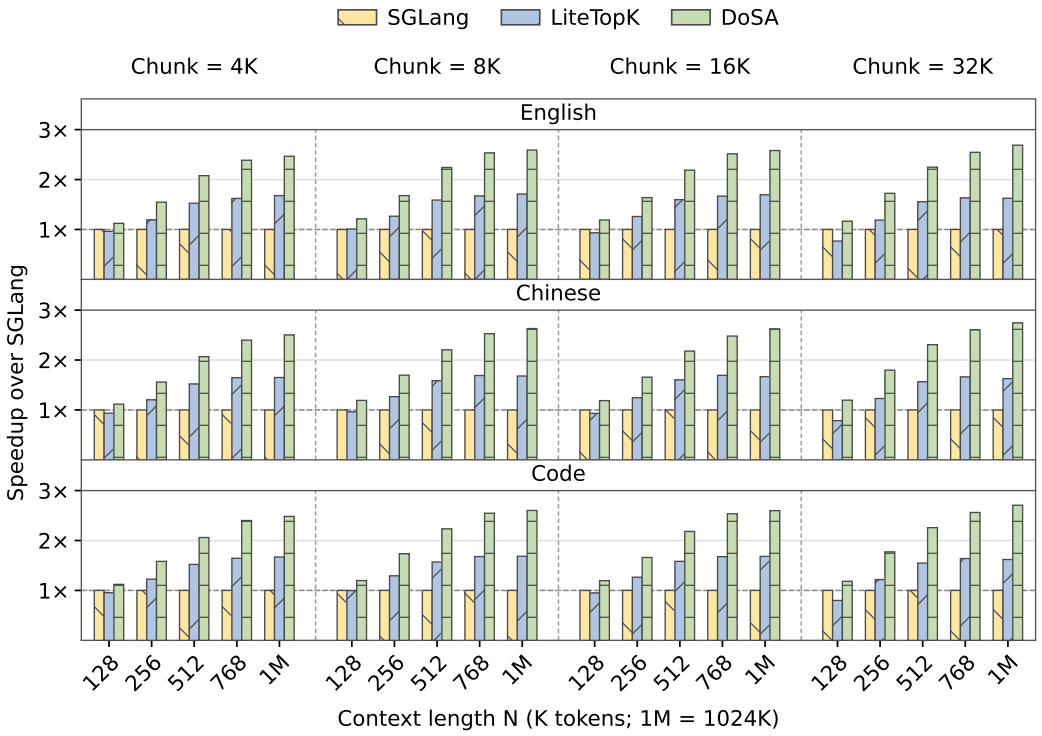


Figure 4: Complete-Indexer speedup over SGLang across context lengths, input domains, and prefill chunk sizes.

4 EXPERIMENTS

DOSA achieves up to $2.74\times$ complete-Indexer speedup and $1.94\times$ TTFT speedup over SGLang. We study how these gains vary with context length, input domain, and prefill chunk size, then isolate the contributions of screening and sparse writeback.

4.1 EXPERIMENTAL SETUP

Models and hardware. We evaluate GLM-5.2 and DeepSeek-V3.2 on NVIDIA GPUs. The models use 32 and 64 indexer heads, respectively, with Top-$K = 2{,}048$. Microbenchmarks run GLM-5.2 on a single GPU. End-to-end inference uses eight-way tensor parallelism, with prefill chunks of 32K tokens for GLM-5.2 and 16K for DeepSeek-V3.2.

Workloads. We use English and Chinese books and code inputs from InfiniteBench (Zhang et al., 2024), supplemented with Project Gutenberg[1] and SQLite source code[2] for long contexts. For each domain, we select samples by tokenized length and extract contiguous contexts between 128K and approximately 1M tokens. The microbenchmarks use prefixes of 1M-token inputs; certificate analyses cover samples throughout this length range.

Baselines. We compare with SGLang v0.5.18 (Zheng et al., 2023) and LiteTopK (Yin et al., 2026). SGLang computes full multihead scores followed by Top-K selection. LiteTopK fuses scoring, candidate filtering, and selection to reduce intermediate memory traffic. DOSA further reduces computation by using certificates to skip unnecessary full-head scoring.

[1] https://www.gutenberg.org/ebooks/100; https://www.gutenberg.org/ebooks/3200

[2] https://www.sqlite.org/amalgamation.html

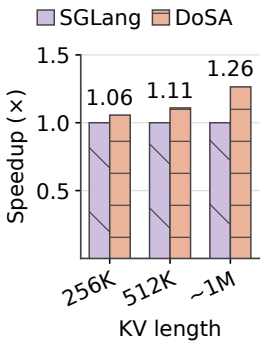


Figure 5: Decoding Indexer speedup over SGLang.

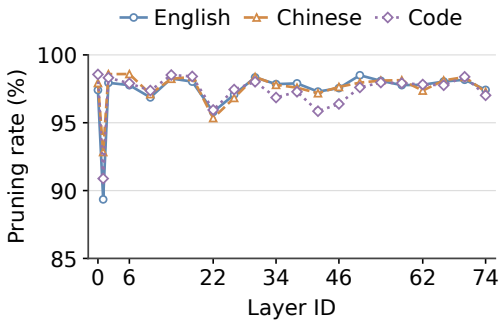


Figure 6: Certificate pruning rates across GLM-5.2 layers, averaged within each input domain.

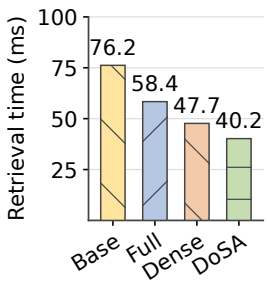


Figure 7: Screening and sparse writeback ablation.

Measurements. We measure complete-Indexer latency with CUDA events after warmup, using activations from model runs. End-to-end performance is measured by time to first token (TTFT) for individual requests, clearing the prefix cache between requests.

4.2 Indexer microbenchmarks

Prefill performance. We vary context length over 128K, 256K, 512K, 768K, and 1M tokens, and prefill chunk size over 4K, 8K, 16K, and 32K. Each configuration replays the final query chunk across all 21 indexer-producing layers, including query/key preparation in the measured latency.

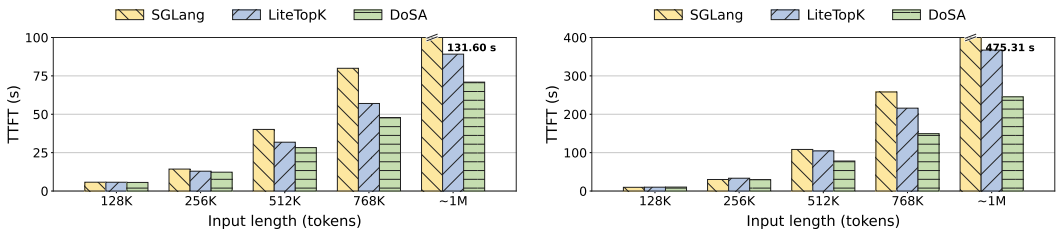
DoSA is the fastest method in all 60 configurations (Figure 4). Speedups over SGLang range from 1.12 to 2.74×, exceeding 2× in every configuration at 512K tokens and above. Against LiteTopK, DoSA achieves 1.17–1.69× speedups. At 1M, the gains reach 2.47–2.74× and 1.47–1.69× over the two baselines, respectively.

Analysis. Context length drives the gain more strongly than query chunk size. For English inputs with 16K queries, speedup rises from 1.19× at 128K to 2.58× at 1M. By comparison, increasing the query chunk eightfold at 1M, from 4K to 32K, changes speedup from 2.47× to 2.69×. Longer contexts expand the key set evaluated by each query while reusing its certificate preparation. Increasing the chunk size instead scales the query batch for all methods. Thus, DoSA gains most from reducing work along the long key dimension, while remaining effective across prefill batching choices.

The comparison with LiteTopK separates this effect from fusion. Fusion reduces score traffic and selection overhead, but leaves the full multihead computation intact. Certified rejection removes this remaining work, explaining why DoSA continues to pull ahead as contexts grow. Appendix D.4 provides the layer-wise retrieval results.

Decoding performance. We build document-analysis requests from HELMET, InfiniteBench, and LongBench v2 (Yen et al., 2024; Zhang et al., 2024; Bai et al., 2024), and code-review requests from SQLite. Each workload fills an approximately 1M-token KV cache with equal-length requests. DoSA accelerates the decoding Indexer by 1.06×, 1.11×, and 1.26× at request lengths of 256K, 512K, and approximately 1M, respectively (Figure 5). Here the gain comes from reducing selection work: recently selected keys provide thresholds for compact selection, benefiting decoding even with few queries per step.

Pruning across layers. We define the pruning rate as $p = 1 - N_{\text{keep}}/N_{\text{screen}}$, where N_{screen} and N_{keep} count query–key pairs before and after certificate screening. Figure 6 shows pruning rates above 95% in 20 of the 21 layers in each domain, with domain averages of 97.2%–97.5%. English, Chinese, and code exhibit similar profiles.



(a) GLM-5.2 (b) DeepSeek-V3.2

Figure 8: Time to first token across input lengths on (a) GLM-5.2 and (b) DeepSeek-V3.2.

The variation follows model depth: all three domains have their lowest pruning rate at layer 1, where 89%–93% of pairs are still rejected, and exceed 95% in every other layer. The shared pattern points to layer-dependent score geometry as a stronger influence than the input domain. In terms of the selection-margin analysis in Appendix B.3, most keys lie sufficiently far below the Top-K boundary that the certificate can reject them despite its slack. Full-head scoring is then concentrated on the small set of candidates that remain competitive.

4.3 End-to-end inference performance

We measure TTFT on the book and code workloads from 128K to approximately 1M tokens, using TP8 and the model-specific prefill chunks described above.

DoSA nearly halves TTFT at 1M tokens on both models (Figure 8). It achieves $1.86\times$ and $1.94\times$ speedups over SGLang on GLM-5.2 and DeepSeek-V3.2, respectively, and $1.26\times$ and $1.50\times$ over LiteTopK. The gains over LiteTopK show that avoiding score computation improves serving latency even after scoring and selection have been fused.

Analysis. The growing Indexer share explains why the end-to-end gain strengthens with context length. With Indexer time fraction f and Indexer speedup s, Amdahl's law gives an overall speedup of $S = (1 - f + f/s)^{-1}$. For GLM-5.2, indexing grows from 23.6% of prefill GPU time at 128K to 73.0% at 1M; DeepSeek-V3.2 reaches 90.8% at 1M (Figure 1). As this fraction increases, accelerating indexing removes a larger share of request latency. Attention and feed-forward layers continue to contribute to request latency, explaining why TTFT speedup is smaller than Indexer speedup. This explains both the increasing gains in Figure 8 and the value of targeting indexing for long-context serving.

4.4 Ablation study

We enable screening and sparse writeback independently: Base uses neither, Full uses sparse writeback alone, Dense uses screening alone, and DoSA combines both. The workload is a SciFact retrieval example from LOFT (Lee et al., 2024), with 16K queries and 896K keys; we average retrieval latency across all 21 indexer-producing layers.

Screening alone reduces latency by 37.4%, and sparse writeback alone by 23.4% (Figure 7). Together, they reduce retrieval latency by 47.3%, yielding a $1.90\times$ speedup over Base. Adding screening to Full saves a further 31.1%, while adding sparse writeback to Dense saves 15.7%.

Analysis. Full avoids dense score writeback but still pays for every head evaluation; its remaining gap to DoSA exposes the computation that fusion alone cannot remove. Dense makes the opposite tradeoff: screening reduces score evaluation, while dense intermediate results continue to consume HBM bandwidth. The additional gain from sparse writeback therefore shows why pruning needs a matching execution strategy to realize its benefit.

The certificate-component ablation further identifies what makes pruning effective. Removing the negative aggregate raises candidate survival from 2.65% to 95.06% (Appendix D.2), nearly eliminating rejection. The negative contribution is therefore essential to obtaining a tight certificate for the signed Indexer score. Appendix D.3 shows how this effect varies across layers.

5 Related work

Sparse attention with lightning indexers. Sparse attention restricts query–key interactions. NSA combines token compression with fine-grained selection (Yuan et al., 2025); MoBA uses expert routing for block attention (Lu et al., 2025); DSA selects tokens with a learned multihead indexer (DeepSeek-AI, 2025). MiniMax-M3 combines block attention with learned indexing (Lai et al., 2026). IndexCache reuses indices across layers (Bai et al., 2026; Z.ai, 2026; Tencent Hy Team, 2026), while SparDA predicts next-layer selections (Fu et al., 2026). HISA uses hierarchical block indexing and token refinement for token-level attention (Xu et al., 2026); DeepSeek-V4.1-Flash and LongCat-2.0 combine hierarchical indexing with cross-layer reuse (DeepSeek-AI, 2026; Meituan LongCat Team, 2026). PIVOT shares proxy scans across query groups (Liu et al., 2026). MISA selects indexer heads with optional full-score reranking (Zhou et al., 2026); Qwen3.8-Flash-Next uses only four query heads in its indexer (Qwen Team, 2026). These approaches need not preserve the original full scorer's Top-K, and reranking cannot recover excluded tokens. DOSA certifies rejection under the original scorer.

Exact GPU selection and fused scoring. Radix selection narrows Top-K candidates through successive digit histograms, without fully sorting the scores (Zhang et al., 2023). GVR uses temporal correlation to predict a decoding threshold, verifies it by counting materialized scores, and refines candidates in shared memory (Cheng et al., 2026). DeepSelect streams materialized score rows through a bounded candidate buffer with a monotonically increasing threshold (Qian et al., 2026). LiteTopK fuses full-head scoring with online candidate filtering, reducing score writeback and selection input (Yin et al., 2026). These methods accelerate Top-K selection but still rely on full-head scores for every key in DSA. DOSA builds on streaming thresholds and compact selection by adding a score upper bound that rejects keys *before* full-head evaluation.

Score bounds in classical document retrieval. MaxScore (Turtle & Flood, 1995) and WAND (Broder et al., 2003) use term-contribution upper bounds and a running Top-K threshold to skip unnecessary document scoring. Block-Max WAND tightens these bounds for individual posting-list blocks (Ding & Suel, 2011). DOSA applies the same pruning principle to the signed multihead scores of DSA. A block-level upper bound must cover every key in the block and can therefore be loose for individual candidates. This motivates tighter, per-key bounds that exploit query-head geometry and activation structure to reject candidates before full-head scoring.

6 Limitations and conclusion

DOSA shows that exact token selection can require less work than computing every token's full score. Signed-head certificates, original-score completion, and sparse writeback turn this observation into conditional computation within the attention indexer. Query geometry determines how much work can be certified unnecessary; the head-budget analysis makes the cost of obtaining that certificate explicit.

The TTFT results show increasing benefits at long contexts, while the workload analysis explains the tradeoff between screening cost and avoided full scoring. DOSA still scans every key, and short contexts or loose bounds can leave insufficient savings to offset screening. Numerical guarantees require conservative evaluation relative to the original scorer; valid tied Top-K sets need not yield identical generated text. The measurements establish computational benefits, not improved task accuracy beyond a model's trained context.

Future work. Training-time regularization of certificate slack could encourage an indexer whose computation is easier to skip safely. Following upper-bound pruning in document retrieval, future work could extend our per-key certificates to key-block bounds using compact summaries. Blocks whose bounds fall below the selection threshold could then be skipped without reading individual key vectors. These directions could reduce both scoring and key-access costs as context grows.

AI USE STATEMENT

We used generative AI tools for manuscript drafting and revision and for drafting initial GPU kernel implementations, which were subsequently revised by the authors. The authors take responsibility for the final manuscript and implementation.

REPRODUCIBILITY STATEMENT

Sections 2 and 3 describe the method and GPU implementation. Appendix A proves certificate validity and exact selection; Appendix B analyzes tightness and computational cost; Appendix C gives numerical conditions and recovery requirements. Section 4 specifies the workloads and metrics, and Appendix D details indexer profiling, certificate ablations, and layer-wise retrieval measurements. Accompanying CPU checks verify algebraic relations and fixed-threshold selection.

REFERENCES

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench v2: Towards Deeper Understanding and Reasoning on Realistic Long-context Multitasks. *arXiv preprint arXiv:2412.15204*, 2024. URL https://arxiv.org/abs/2412.15204.

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.

Andrei Z. Broder, David Carmel, Michael Herscovici, Aya Soffer, and Jason Zien. Efficient query evaluation using a two-level retrieval process. In *Proceedings of the Twelfth International Conference on Information and Knowledge Management*, pp. 426–434. ACM, 2003. doi: 10.1145/956863.956944.

Long Cheng, Ritchie Zhao, Timmy Liu, Mindy Li, Xianjie Qiao, Kefeng Duan, Yu-Jung Chen, Xiaoming Chen, Bita Darvish Rouhani, and June Yang. Guess-Verify-Refine: Data-Aware Top-K for Sparse-Attention Decoding on Blackwell via Temporal Correlation. *arXiv preprint arXiv:2604.22312*, 2026. URL https://arxiv.org/abs/2604.22312.

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-V4.1-Flash: Pushing the limits of KV cache compression. Technical report, DeepSeek-AI, 2026. URL https://huggingface.co/deepseek-ai/DeepSeek-V4.1-Flash/blob/main/DeepSeek_V41_Tech_Report.pdf.

Shuai Ding and Torsten Suel. Faster top-k document retrieval using block-max indexes. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 993–1002. ACM, 2011. doi: 10.1145/2009916.2010048. URL https://doi.org/10.1145/2009916.2010048.

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://www.jmlr.org/papers/v23/21-0998.html.

Yaosheng Fu, Guangxuan Xiao, Xin Dong, Song Han, and Oreste Villa. SparDA: Sparse Decoupled Attention for Efficient Long-Context LLM Inference. *arXiv preprint arXiv:2606.04511*, 2026. URL https://arxiv.org/abs/2606.04511.

Xunhao Lai, Weiqi Xu, Yufeng Yang, Qiaorui Chen, Yang Xu, Lunbin Zeng, Xiaolong Li, Haohai Sun, Haichao Zhu, Vito Zhang, Jinkai Hu, Jiayao Li, Rui Gao, Zekun Li, Songquan Zhu, Jingkai Zhou, and Pengyu Zhao. MiniMax Sparse Attention. *arXiv preprint arXiv:2606.13392*, 2026. URL https://arxiv.org/abs/2606.13392.

Jinhyuk Lee, Anthony Chen, Zhuyun Dai, Dheeru Dua, Devendra Singh Sachan, Michael Boratko, Yi Luan, Sébastien M. R. Arnold, Vincent Perot, Siddharth Dalmia, Hexiang Hu, Xudong Lin, Panupong Pasupat, Aida Amini, Jeremy R. Cole, Sebastian Riedel, Iftekhar Naim, Ming-Wei Chang, and Kelvin Guu. Can Long-Context Language Models Subsume Retrieval, RAG, SQL, and More? *arXiv preprint arXiv:2406.13121*, 2024. URL https://arxiv.org/abs/2406.13121.

Hong Liu, Yuan Cheng, Lin Niu, Yi Su, Yufei Xue, Anmin Liu, Guanghua Yu, and Jianchen Zhu. PIVOT: Efficient Query-Group Indexing for Token-Level Sparse Attention. *arXiv preprint arXiv:2607.24593*, 2026. URL https://arxiv.org/abs/2607.24593.

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://arxiv.org/abs/2502.13189.

Meituan LongCat Team. LongCat-2.0. Official model card, 2026. URL https://huggingface.co/meituan-longcat/LongCat-2.0.

Yi Qian, Shengyu Liu, and Yichen Li. DeepSelect: High-performance TopK kernels for DeepSeek sparse attention and sampling. GitHub repository and algorithm documentation, 2026. URL https://github.com/deepseek-ai/DeepSelect.

Qwen Team. On the design of Qwen3.8-Next architecture: Evaluation, efficiency, and training stability. Technical report, Qwen, 2026. URL https://github.com/QwenLM/Qwen3.8-Flash-Next/blob/main/tech_report.pdf.

Tencent Hy Team. Hy4-preview. Official model card, 2026. URL https://huggingface.co/tencent/Hy4-preview.

Howard Turtle and James Flood. Query evaluation: Strategies and optimizations. *Information Processing & Management*, 31(6):831–850, 1995. doi: 10.1016/0306-4573(95)00020-H.

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://arxiv.org/abs/2603.28458.

Howard Yen, Tianyu Gao, Minmin Hou, Ke Ding, Daniel Fleischer, Peter Izsak, Moshe Wasserblat, and Danqi Chen. HELMET: How to Evaluate Long-Context Language Models Effectively and Thoroughly. *arXiv preprint arXiv:2410.02694*, 2024. URL https://arxiv.org/abs/2410.02694.

Ziqi Yin, Jianyang Gao, Peiqi Yin, Jiangneng Li, and Gao Cong. LiteTopK: Exploiting the Curse of Dimensionality for a Fused Indexer-TopK Kernel in Long-Context Sparse Attention. *arXiv preprint arXiv:2607.11976*, 2026. URL https://arxiv.org/abs/2607.11976.

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.

Z.ai. GLM-5.3. Official model documentation, 2026. URL https://github.com/zai-org/GLM-5.

Jingrong Zhang, Akira Naruse, Xipeng Li, and Yong Wang. Parallel Top-K algorithms on GPU: A comprehensive study and new methods. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–13. ACM, 2023. doi: 10.1145/3581784.3607062. URL https://doi.org/10.1145/3581784.3607062.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Khai Hao, Xu Han, Zhen Leng Thai, Shuo Wang, Zhiyuan Liu, and Maosong Sun. ∞Bench: Extending Long Context Evaluation Beyond 100K Tokens. *arXiv preprint arXiv:2402.13718*, 2024. URL https://arxiv.org/abs/2402.13718.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark Barrett, and Ying Sheng. SGLang: Efficient Execution of Structured Language Model Programs. *arXiv preprint arXiv:2312.07104*, 2023. URL https://arxiv.org/abs/2312.07104.

Ruijie Zhou, Fanxu Meng, Yufei Xu, Tongxuan Liu, Guangming Lu, Muhan Zhang, and Wenjie Pei. MISA: Mixture of Indexer Sparse Attention for Long-Context LLM Inference. *arXiv preprint arXiv:2605.07363*, 2026. URL https://arxiv.org/abs/2605.07363.

A CERTIFICATE VALIDITY AND EXACT SELECTION

We establish the score bounds in real arithmetic and prove exact selection for a fixed query over its visible keys. Numerical conditions are given in Appendix C.

A.1 VALIDITY OF THE SCORE CERTIFICATE

Proposition 1 (Positive-head projection bound). *For any $\beta_h \geq 0$ and decomposition $\mathbf{q}_h = \beta_h \mathbf{q}_j + \mathbf{d}_h$,*

$$[\mathbf{q}_h^\top \mathbf{k}]_+ \leq \beta_h [\mathbf{q}_j^\top \mathbf{k}]_+ + \|\mathbf{d}_h\|_2 \, \|\mathbf{k}\|_2 \, .$$

Proof. For real a, b and $c \geq 0$, $[a+b]_+ \leq [a]_+ + [b]_+$ and $[ca]_+ = c[a]_+$. Hence

$$[\beta_h \mathbf{q}_j^\top \mathbf{k} + \mathbf{d}_h^\top \mathbf{k}]_+ \leq \beta_h [\mathbf{q}_j^\top \mathbf{k}]_+ + [\mathbf{d}_h^\top \mathbf{k}]_+ \leq \beta_h [\mathbf{q}_j^\top \mathbf{k}]_+ + \|\mathbf{d}_h\|_2 \, \|\mathbf{k}\|_2 \, .$$

Multiplying by $w_h > 0$ and summing over P gives the positive part of Equation (9). Any finite nonnegative coefficient is valid if its residual is bounded conservatively. □

Proposition 2 (Negative-head support bound). *For any $A \subseteq N$, $[\mathbf{a}_A^\top \mathbf{k}]_+ \leq \sum_{h \in N} |w_h| [\mathbf{q}_h^\top \mathbf{k}]_+$.*

Proof. For $c_h \geq 0$, ReLU subadditivity gives $[\sum_{h \in A} c_h x_h]_+ \leq \sum_{h \in A} c_h [x_h]_+ \leq \sum_{h \in N} c_h [x_h]_+$. Substitute $c_h = |w_h|$ and $x_h = \mathbf{q}_h^\top \mathbf{k}$. The result holds pointwise for any data-dependent support, including $A = \varnothing$, when the left-hand side is zero. □

Together, these bounds establish $s_i \leq U_i$ in Equation (9). The following grouped and joint bounds can tighten its positive residual allowance.

Grouped positive residuals. Partition the coordinates into G disjoint groups. In real arithmetic, let $E_{P,g} = \sum_{h \in P} w_h \left\| \mathbf{d}_h^{(g)} \right\|_2$. Then

$$\sum_{h \in P} w_h [\mathbf{d}_h^\top \mathbf{k}_i]_+ \leq \sum_{g=1}^{G} E_{P,g} \left\| \mathbf{k}_i^{(g)} \right\|_2 \tag{10}$$

$$\leq \left(\sum_g E_{P,g}^2 \right)^{1/2} \|\mathbf{k}_i\|_2 \, . \tag{11}$$

Minkowski's inequality gives

$$\left(\sum_g E_{P,g}^2 \right)^{1/2} = \left\| \sum_{h \in P} w_h \left(\left\| \mathbf{d}_h^{(1)} \right\|_2, \ldots, \left\| \mathbf{d}_h^{(G)} \right\|_2 \right) \right\|_2 \leq \sum_{h \in P} w_h \|\mathbf{d}_h\|_2 = E_P.$$

This tightens the residual coefficient with one RMS value per key.

Joint positive residuals. The grouped bound can also exploit dependence among residual directions. For one coordinate group, write $\mathbf{u}_h = w_h \mathbf{d}_h^{(g)}$ and define

$$B_g = \tfrac{1}{2} \left(\left\| \sum_{h \in P} \mathbf{u}_h \right\|_2 + \sqrt{\sum_{h,\ell \in P} |\mathbf{u}_h^\top \mathbf{u}_\ell|} \right) . \tag{12}$$

Proposition 3 (Joint residual allowance). *For every $\mathbf{k}^{(g)}$, $\sum_{h \in P} [\mathbf{u}_h^\top \mathbf{k}^{(g)}]_+ \leq B_g \left\| \mathbf{k}^{(g)} \right\|_2$, and $B_g \leq \sum_{h \in P} \|\mathbf{u}_h\|_2 = E_{P,g}$.*

Proof. Using $[x]_+ = (x + |x|)/2$ gives

$$\sum_h [\mathbf{u}_h^\top \mathbf{k}^{(g)}]_+ = \tfrac{1}{2} \left[\left(\sum_h \mathbf{u}_h \right)^\top \mathbf{k}^{(g)} + \sum_h |\mathbf{u}_h^\top \mathbf{k}^{(g)}| \right] .$$

The absolute-value sum equals $\max_{\sigma_h \in \{-1,1\}} (\sum_h \sigma_h \mathbf{u}_h)^\top \mathbf{k}^{(g)}$. For every sign vector, $\|\sum_h \sigma_h \mathbf{u}_h\|_2^2 \le \sum_{h,\ell} |\mathbf{u}_h^\top \mathbf{u}_\ell|$. Cauchy–Schwarz yields the first claim. The triangle inequality bounds the center norm by $\sum_h \|\mathbf{u}_h\|_2$, and $\sum_{h,\ell} |\mathbf{u}_h^\top \mathbf{u}_\ell| \le (\sum_h \|\mathbf{u}_h\|_2)^2$ proves the second. □

ReLU subadditivity and Cauchy–Schwarz combine groups into $\sqrt{\sum_g B_g^2}\, \|\mathbf{k}\|_2$. A single group proves Equation (7). The bound needs no additional key summary, but its query-side cost is quadratic in the number of omitted positive heads.

A.2 Fixed-threshold algorithm and exact Top-K selection

Assume finite scores and $|\Omega| \ge K$. Top-K returns K distinct token IDs; identical tied IDs require a common total order. A seed set of distinct visible keys supplies a threshold that remains fixed during screening and refinement.

Proposition 4 (Subset threshold)**.** *For any $\mathcal{S} \subseteq \Omega$ with $|\mathcal{S}| \ge K$, $\tau_K(\mathcal{S}) \le \tau_K(\Omega)$.*

Proof. At least K distinct elements of $\mathcal{S}$, hence of Ω, have score at least $\tau_K(\mathcal{S})$. The full-domain Kth score can therefore be no smaller. □

Algorithm 1 DOSA with a fixed seed threshold

1. Choose distinct visible seed keys $\mathcal{S} \subseteq \Omega$, $|\mathcal{S}| \ge K$.
2. Score $\mathcal{S}$ with the original scorer; $\tau \leftarrow \tau_K(\mathcal{S})$.
3. Build R, anchors, A, and the score bound in Equation (9).
4. $V \leftarrow \{i \in \Omega \setminus \mathcal{S} : U_i \ge \tau\}$.
5. Score every $i \in V$ using the original H-head scorer.
6. $C \leftarrow \{i \in V : s_i \ge \tau\}$.
7. Return $\mathrm{TopK}(\mathcal{S} \cup C)$.

Both filters retain equality. Final selection uses original scores and the common tie order when identical tied IDs are required.

Theorem 1 (Exact Top-K selection)**.** *In Algorithm 1, let $\tau = \tau_K(\mathcal{S})$ come from at least K distinct visible keys scored by the original scorer. Suppose every key rejected during screening satisfies $s_i \le U_i < \tau$, and no live candidate is lost. With a common total tie order, the algorithm returns the same Top-K as full-domain scoring. Without a prescribed tie order, it returns a valid Top-K set under Equation* (1)*.*

Proof of Theorem 1. Consider any key $i \notin \mathcal{S} \cup C$. If screening rejects it, then $s_i \le U_i < \tau$. Otherwise it belongs to $V \setminus C$ and its original score satisfies $s_i < \tau$. In both cases, the subset-threshold proposition gives

$$s_i < \tau_K(\mathcal{S}) \le \tau_K(\Omega).$$

Thus no discarded key can belong to the full-domain Top-K, including under any tie order. Since $\mathcal{S} \cup C$ contains at least K distinct keys and every possible winner, applying the same total order to its original scores returns exactly $\mathrm{TopK}(\Omega)$. Without a prescribed tie order, selection includes every score strictly above the global Kth score and fills the remaining places with distinct keys at that score. Identical tied IDs are not required in that case. □

Candidate-storage and recovery requirements that enforce the no-loss condition are given in Appendix C.3.

B Certificate Tightness and Computational Cost

We analyze bound slack, selection margins, and screening cost in finite real arithmetic. These determine efficiency separately from correctness.

B.1 Minimum uniform allowance for a single anchor

Proposition 5 (Optimal single-anchor allowance)**.** *For vectors $\mathbf{q}, \mathbf{p} \in \mathbb{R}^D$, consider all pairs $\beta, c \geq 0$ satisfying*

$$[\mathbf{q}^\top \mathbf{k}]_+ \leq \beta [\mathbf{p}^\top \mathbf{k}]_+ + c \|\mathbf{k}\|_2 \quad \textit{for every } \mathbf{k} \in \mathbb{R}^D. \tag{13}$$

The smallest achievable coefficient is $c^* = \min_{\beta \geq 0} \|\mathbf{q} - \beta \mathbf{p}\|_2$, the distance from $\mathbf{q}$ to the nonnegative ray generated by $\mathbf{p}$. For $\mathbf{p} \neq 0$, it is attained by

$$\beta^* = \frac{[\mathbf{q}^\top \mathbf{p}]_+}{\|\mathbf{p}\|_2^2}, \qquad c^* = \|\mathbf{q} - \beta^* \mathbf{p}\|_2 .$$

For $\mathbf{p} = 0$, take $\beta^* = 0$ and $c^* = \|\mathbf{q}\|_2$.

Proof of Proposition 5. The positive-head projection bound already establishes feasibility of (β^*, c^*). We prove that no choice of β permits a smaller c. The case $\mathbf{q} = 0$ is immediate. If $\mathbf{q}^\top \mathbf{p} \leq 0$, substitute $\mathbf{k} = \mathbf{q}/\|\mathbf{q}\|_2$ into Equation (13). Its anchor term vanishes for every $\beta \geq 0$, giving $c \geq \|\mathbf{q}\|_2 = c^*$. This includes a zero anchor. Otherwise let $\mathbf{d} = \mathbf{q} - \beta^* \mathbf{p}$. The projection satisfies $\mathbf{d}^\top \mathbf{p} = 0$. When $\mathbf{d} \neq 0$, substituting $\mathbf{k} = \mathbf{d}/\|\mathbf{d}\|_2$ again makes the anchor term vanish, while $[\mathbf{q}^\top \mathbf{k}]_+ = \|\mathbf{d}\|_2$. Thus $c \geq c^*$ for every β. If $\mathbf{d} = 0$, feasibility with $c = 0$ attains the trivial lower bound. □

Minimizing c^* over retained anchors gives Equation (4). This optimizes the uniform norm allowance, not the complete bound pointwise or the retained-head set. Rounded coefficients remain valid when their actual residuals are conservatively bounded.

B.2 Certificate slack and activation-aware head selection

For one key, abbreviate $x_h = \mathbf{q}_h^\top \mathbf{k}$ and $c_h = |w_h|$. Let

$$\begin{aligned}
B_h(\mathbf{k}) &= \beta_h [\mathbf{q}_{j_h^\star}^\top \mathbf{k}]_+ + \|\mathbf{d}_h\|_2 \|\mathbf{k}\|_2 , \\
\delta_P(\mathbf{k}) &= \sum_{h \in P} w_h \big(B_h(\mathbf{k}) - [x_h]_+\big), \\
Z_A^+(\mathbf{k}) &= \sum_{h \in A} c_h [x_h]_+, \qquad Z_A^-(\mathbf{k}) = \sum_{h \in A} c_h [-x_h]_+ .
\end{aligned}$$

Proposition 6 (Exact certificate slack)**.** *With the notation above, the bound in Equation* (9) *has three nonnegative slack components:*

$$\boxed{U(\mathbf{k}) - s(\mathbf{k}) = \underbrace{\delta_P(\mathbf{k})}_{\textit{positive projection}} + \underbrace{\sum_{h \in N \setminus A} c_h [x_h]_+}_{\textit{missed active negatives}} + \underbrace{\min\{Z_A^+(\mathbf{k}), Z_A^-(\mathbf{k})\}}_{\textit{aggregate cancellation}} .} \tag{14}$$

Proof of Proposition 6. The retained contributions cancel in $U - s$. The positive remainder is $\delta_P \geq 0$ by Equation (5). The negative remainder is

$$\begin{aligned}
\delta_N(\mathbf{k}) &= \sum_{h \in N} c_h [x_h]_+ - [Z_A^+ - Z_A^-]_+ \\
&= \sum_{h \in N \setminus A} c_h [x_h]_+ + \min\{Z_A^+, Z_A^-\},
\end{aligned}$$

where the last equality uses $z - [z - y]_+ = \min(z, y)$ for $z, y \geq 0$. This proves Equation (14) and nonnegativity of all its components. □

Let $M(\mathbf{k}) = \{h \in N : x_h > 0\}$ be the active omitted negative heads. Choosing $A = M(\mathbf{k})$ makes both negative slack terms zero; including inactive heads can instead introduce cancellation. The positive headwise decomposition above uses the triangle allowance E_P. Replacing it with a valid joint allowance preserves nonnegative slack but changes its positive term.

Support and retained-head selection. For any support, the negative slack obeys

$$0 \le \delta_N(\mathbf{k}) \le \sum_{h \in A \triangle M(\mathbf{k})} c_h |\mathbf{q}_h^\top \mathbf{k}| \le \|\mathbf{k}\|_2 \sum_{h \in A \triangle M(\mathbf{k})} a_h, \qquad a_h = c_h \|\mathbf{q}_h\|_2 , \tag{15}$$

where $\triangle$ denotes symmetric difference. The first inequality bounds aggregate cancellation by Z_A^-; the second is Cauchy–Schwarz.

For nonzero sample keys, let p_h be the observed activation frequency. Averaging the last expression in Equation (15) yields the separable surrogate

$$J(A) = \sum_{h \in A} a_h(1 - p_h) + \sum_{h \in N \setminus A} a_h p_h. \tag{16}$$

If the masks are the signs of the same real dot products, $J(A)$ upper-bounds the sample mean of $\delta_N(\mathbf{k})/\|\mathbf{k}\|_2$. Each head independently contributes either $a_h(1-p_h)$ or $a_h p_h$, so componentwise majority, $A^* = \{h : p_h \ge 1/2\}$, minimizes J. The minimum is $\sum_h a_h \min(p_h, 1-p_h)$.

If all positive heads are retained, evaluating b additional negative heads removes their surrogate terms. Retaining the b largest $a_h \min(p_h, 1-p_h)$ therefore minimizes the remaining surrogate jointly with majority support. This optimizes the sampled disagreement bound, not survivor count or latency. Rounded activation signs near zero can affect the surrogate interpretation, but any resulting support remains safe.

B.3 Selection margins and screening cost

Let τ^* be the full-domain Kth score, $\Delta_i = \tau^* - s_i$ a key's selection margin, $b = \tau^* - \tau \ge 0$ the threshold deficit, and $\delta_i = U_i - s_i \ge 0$ the bound slack. Then

$$\boxed{U_i < \tau \iff \Delta_i > \delta_i + b.} \tag{17}$$

A stronger seed reduces b; a tighter certificate reduces δ_i. If $\delta_i \le \epsilon$, survivors lie within $\{i : s_i \ge \tau^* - \epsilon - b\}$. This boundary mass need not be small, and tied keys survive automatically. The same relation holds for computed scores s_i^{ref} and valid numerical bounds U_i^{num}; keys without valid bounds are retained separately.

Arithmetic cost and head budget. The survivor count determines whether screening saves arithmetic. Let N_0 keys be fully scored during initialization, N_s be screened, and M survive screening. With d_a auxiliary rows, logical dot-product counts are approximately

$$D_{\mathrm{base}} = H(N_0 + N_s), \tag{18}$$

$$D_{\mathrm{DoSA}} = HN_0 + (r + d_a)N_s + HM. \tag{19}$$

Writing $f = N_0/(N_0 + N_s)$ and $m = M/N_s$, normalized work is

$$\frac{D_{\mathrm{DoSA}}}{D_{\mathrm{base}}} \approx f + (1 - f)\left(\frac{r + d_a}{H} + m\right). \tag{20}$$

Screening saves logical dot products only when $m < 1 - (r + d_a)/H$. More precisely, let $p(r) = 1 - m(r)$ be rejection at budget r. Within this logical-work model, for fixed initialization and auxiliary-row count,

$$D(r + \Delta r) < D(r) \iff p(r + \Delta r) - p(r) > \frac{\Delta r}{H}. \tag{21}$$

An additional head pays off only if it avoids enough full rescoring. If the auxiliary-row count changes, replace Δr by $\Delta r + \Delta d_a$; changed anchors or supports need not improve rejection monotonically. This model excludes planning, key summaries, Tensor Core padding, selection, and recovery. Every key is still scanned, and screening adds work if all keys survive.

C Numerical Conditions and Recovery

This section gives sufficient conditions for preserving exact selection under the original scorer's arithmetic, together with candidate-storage and recovery requirements.

C.1 Reference scores and the FP8 certificate

Let s_i^{ref} be the original indexer's computed score. Numerical dominance $s_i^{\mathrm{ref}} \le U_i^{\mathrm{num}}$ must be established separately from $s_i \le U_i$. The tie-order requirements of Theorem 1 also apply to computed scores, and equality survives pruning.

Theorem 1 applies with s_i^{ref} and U_i^{num} when the seed threshold and candidate scores use the same original scorer and every key rejected during screening satisfies $s_i^{\mathrm{ref}} \le U_i^{\mathrm{num}} < \tau$. Failure handling is specified in Appendix C.3.

Write an FP8 key as $\mathbf{k}_i = \alpha_i \bar{\mathbf{k}}_i$, where $\alpha_i > 0$ and $\bar{\mathbf{k}}_i$ is its D-dimensional payload. In the base form of Equation (9), two auxiliary rows approximate the negative aggregate:

$$\mathbf{a}_A = \mathbf{v}_0 + \mathbf{v}_1 + \mathbf{e}_A. \tag{22}$$

Their dot products are added before ReLU. With stored folded weights ω_j, the base screening expression in payload units is

$$\widehat{U}_i = \mathrm{fl}_{\mathrm{screen}}\left(\sum_{j \in R} \omega_j [\mathbf{q}_j^\top \bar{\mathbf{k}}_i]_+ - [(\mathbf{v}_0 + \mathbf{v}_1)^\top \bar{\mathbf{k}}_i]_+ \right). \tag{23}$$

Absolute-value form. When every positive-weight head is retained, $P = \varnothing$. Set $\mathbf{V} = \mathbf{a}_A - \frac{1}{2}\sum_{j \in R} w_j \mathbf{q}_j$. Using $[x]_+ = (x + |x|)/2$,

$$\begin{aligned} U_{\mathrm{abs}}(\mathbf{k}) &= \tfrac{1}{2} \sum_{j \in R} w_j |\mathbf{q}_j^\top \mathbf{k}| - \mathbf{V}^\top \mathbf{k} \\ &= S_R(\mathbf{k}) - \mathbf{a}_A^\top \mathbf{k} \ge S_R(\mathbf{k}) - [\mathbf{a}_A^\top \mathbf{k}]_+ \ge s(\mathbf{k}). \end{aligned} \tag{24}$$

The additional slack relative to Equation (9) is $[-\mathbf{a}_A^\top \mathbf{k}]_+$. The 32-head implementation uses this form for $r = 14, 16$; the $r = 20$ path retains the base form with positive projection. Encoding $\mathbf{V} = \mathbf{v}_0 + \mathbf{v}_1 + \mathbf{e}_V$ with two scaled FP8 rows gives

$$\widehat{U}_i = \mathrm{fl}_{\mathrm{screen}}\left(\tfrac{1}{2} \sum_{j \in R} w_j |\mathbf{q}_j^\top \bar{\mathbf{k}}_i| - (\mathbf{v}_0 + \mathbf{v}_1)^\top \bar{\mathbf{k}}_i \right). \tag{25}$$

Both forms use $r+2$ logical dot products; survivors receive the original H-head scores. The numerical allowances cover auxiliary-vector representation, weight formation, and arithmetic errors, together with any omitted positive residuals.

Let $\rho_i \ge \left\|\bar{\mathbf{k}}_i\right\|_2 / \sqrt{D}$ be a conservative key RMS summary. A numerical certificate has the form

$$U_i^{\mathrm{num}} = \mathrm{Eval}_{\mathrm{up}}\left[\alpha_i \big(\widehat{U}_i + \rho_i C + \varepsilon_{\mathrm{raw}} \big) + \varepsilon_{\mathrm{out}} \right]. \tag{26}$$

The nonnegative allowances C, $\varepsilon_{\mathrm{raw}}$, and $\varepsilon_{\mathrm{out}}$ satisfy Appendix C.2. The operator $\mathrm{Eval}_{\mathrm{up}}$ encloses the expression upward through directed rounding or explicit error bounds; round-to-nearest alone is insufficient. These conditions ensure $s_i^{\mathrm{ref}} \le U_i^{\mathrm{num}}$ for either screening expression.

C.2 Error allowances and conservative key summaries

Let E_g^+ be an upper enclosure of a valid positive residual allowance for group g, using the grouped triangle or joint bound in Appendix A.1. Residual formation and group norms must be enclosed upward; joint bounds additionally require conservative weighted residuals, sums, Gram entries, and final norms. Suppose a nonnegative coordinate envelope $\mathbf{J}$ and absolute allowances satisfy

$$s_i^{\mathrm{ref}} \le \alpha_i \left(\widehat{U}_i + \sum_g E_g^+ \left\| \bar{\mathbf{k}}_i^{(g)} \right\|_2 + \sum_d J_d |\bar{k}_{i,d}| + \varepsilon_{\mathrm{raw}} \right) + \varepsilon_{\mathrm{out}}. \tag{27}$$

The envelope covers auxiliary-vector reconstruction, weight formation, screening arithmetic, and the original scorer's arithmetic. In the base form, aggregate error contributes $|e_{A,d}|$ and folded-weight errors contribute $\sum_{j\in R} |\widetilde{w}_j - \omega_j||q_{j,d}|$ to J_d, by ReLU's Lipschitz property. In the absolute-value form, $E_g^+ = 0$ and the auxiliary-vector contribution is $|e_{V,d}|$; the absolute-value and linear terms admit the same coordinate-envelope treatment. Underflow and other absolute errors enter $\varepsilon_{\mathrm{raw}}$ or $\varepsilon_{\mathrm{out}}$ according to whether they occur before or after key scaling.

Applying Cauchy–Schwarz within groups and then across groups shows that a sufficient RMS coefficient is any conservative enclosure

$$C \geq \sqrt{D}\left(\sum_g \left[E_g^+ + \left\|\mathbf{J}^{(g)}\right\|_2\right]^2\right)^{1/2}. \tag{28}$$

Together with conservative ρ_i, this establishes Equation (26). In the absolute-value paths, directed FP64 intervals enclose $\mathbf{V}$ and its reconstruction residual. The planner combines these with screening and reference-arithmetic allowances and rounds C outward to FP32; the kernels evaluate the correction and positive key scaling with upward rounding.

Conservative key RMS. The stored summary must satisfy $\rho_i\sqrt{D} \geq \left\|\bar{\mathbf{k}}_i\right\|_2$. This requires a conservative norm computation and upward coverage in the stored summary format; a nearest-rounded RMS is insufficient. Nonfinite payloads or a failure to establish coverage invalidate the certificate.

C.3 Candidate storage and full-domain recovery

The final output contains K entries, but candidates awaiting full scoring or selection may be far more numerous. A buffer capacity C_{buf} does not limit how many keys satisfy $U_i \geq \tau$: if all scores equal τ, every key passes even with exact bounds. A full buffer must be processed and reduced through exact Top-K merging of original scores, extended, or handled by full-domain recovery. Dropping overflow candidates can lose required Top-K IDs. Chunked processing must preserve the selection and tie-order contract of Algorithm 1.

Invalid scales, RMS summaries, or nonfinite bounds prevent pruning. Query-level errors remain flagged until full-domain recovery, which evaluates the original scorer and selector over all visible keys, including those discarded before error detection. Re-selecting from an incomplete buffer is insufficient. Exceptional values follow the backend's policy; the finite-score theorem does not define NaN semantics.

D Additional Experimental Analysis

D.1 Indexer profiling methodology

Figure 1 uses real model weights and text with DoSA and LiteTopK disabled. Each model–length pair contains two independent requests with a cleared prefix cache. Time is the mean over requests and tensor-parallel ranks of Nsight Systems GPU intervals, cross-checked with CUDA events. The indexer interval includes query/key preparation, key-cache access, scoring, and selection; the denominator is the complete model prefill GPU interval. These quantities exclude CPU tokenization and request transport and are distinct from TTFT.

GLM-5.2 uses 32K prefill chunks and DeepSeek-V3.2 uses 16K chunks, both with TP8. FLOPs count causal logical matrix work, including active experts and sparse attention, with an FMA counted as two operations and no duplication for tensor parallelism. Top-K comparisons, elementwise operations, and padding are not matrix FLOPs. Here 1K denotes 1,024 tokens, 375K denotes 384,000 tokens, and approximately 1M denotes 1,048,569 tokens. DeepSeek measurements beyond its declared context support a computational scaling observation, not a claim about task accuracy.

D.2 Certificate component ablation

We isolate the contributions of the certificate components on the per-domain sample sets described in Section 4. For each of the 21 indexer-producing layers of GLM-5.2, we replay the last 4K queries

with the same initial candidate pool. We compare DoSA with four variants: aggregating all omitted negative heads, removing the negative aggregate, selecting retained negative heads by strength alone, and removing positive-head projection. Each variant is timed ten times per input and layer.

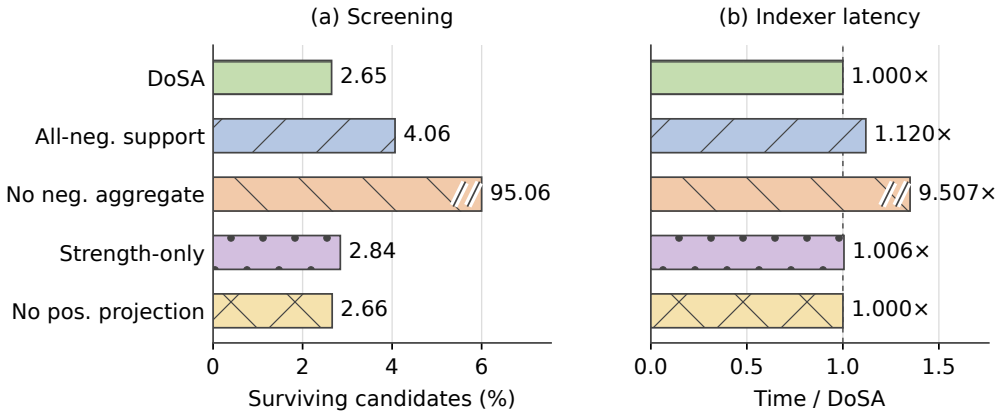


Figure 9: Effects of certificate components on screening and Indexer latency. Bars show the mean candidate survival rate and geometric mean latency relative to DoSA. Breaks mark bars beyond the displayed range; annotations give their measured values.

Figure 9 shows that removing negative aggregation raises candidate survival from 2.65% to 95.06% and increases Indexer time by $9.51\times$. These timings include recovery when the candidate pool fills. Aggregating all negative heads also weakens filtering: survival rises to 4.06% and latency increases by 12.0%, supporting the use of activation-informed support. Strength-only head selection and removal of positive-head projection have much smaller aggregate effects in this workload. Positive-head projection is required by only 0.42% of the evaluated query-layer rows. Appendix D.3 breaks down these effects by layer.

D.3 Layer-wise certificate behavior

Figure 10 decomposes the experiment in Appendix D.2 by layer, aggregating over the same per-domain sample sets. Removing the negative aggregate leaves nearly all candidates alive in most layers. The cost of aggregating all negative heads is most pronounced in early layers: at layer 2, its relative Indexer latency reaches $3.41\times$.

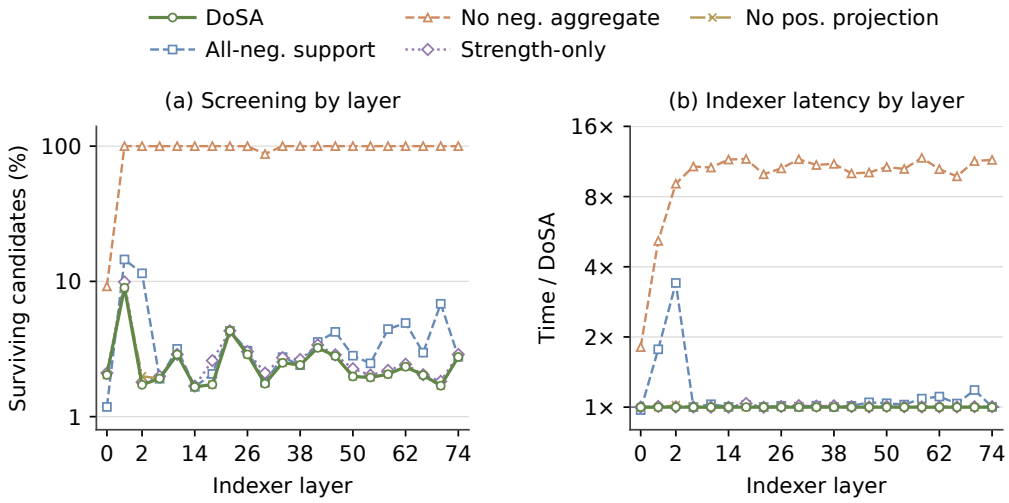


Figure 10: Layer-wise effects of certificate components on candidate survival and Indexer latency. Each layer aggregates the same sample sets. Both vertical axes use logarithmic scales.

D.4 Layer-wise retrieval speedups

For the GLM-5.2 sample sets used in the certificate analysis, we replay the final 32K queries across all 21 indexer-producing layers (252 cells). Figures 11 and 12 divide baseline median latency by DoSA median latency over 12 replays per method and cell. Timings cover scoring and selection on prepared queries and keys, excluding query/key preparation. The heatmaps resolve variation across inputs and layers.

Measurement scope. The matrix retains 247 cells from the original run and replaces five cells with matched measurements of all three methods after a candidate-storage capacity repair. The remeasured cells are EN01 at layer 26 and EN02, EN04, ZH04, and CD04 at layer 1. The repair preserves the scoring and selection rules, four-anchor initialization, and 16K internal query scratch. At approximately 1M-key capacity, it reserves 1,024 candidate-ID pages per query and 114,688 output records, adding 9.94 GiB of workspace relative to the original allocation. The full matrix was not rerun after the repair.

Within these measured cells, DoSA is faster in 251 of 252 comparisons (99.6%) against SGLang and 250 of 252 (99.2%) against LiteTopK. All 21 layers achieve at least $2\times$ speedup over SGLang on one or more inputs, with gains reaching $3.74\times$; speedups over LiteTopK reach $2.08\times$. The three slower comparisons add 0.7–9.0% latency and are retained in the figures. Across the input samples, every layer has a median speedup above $1\times$ against both baselines.

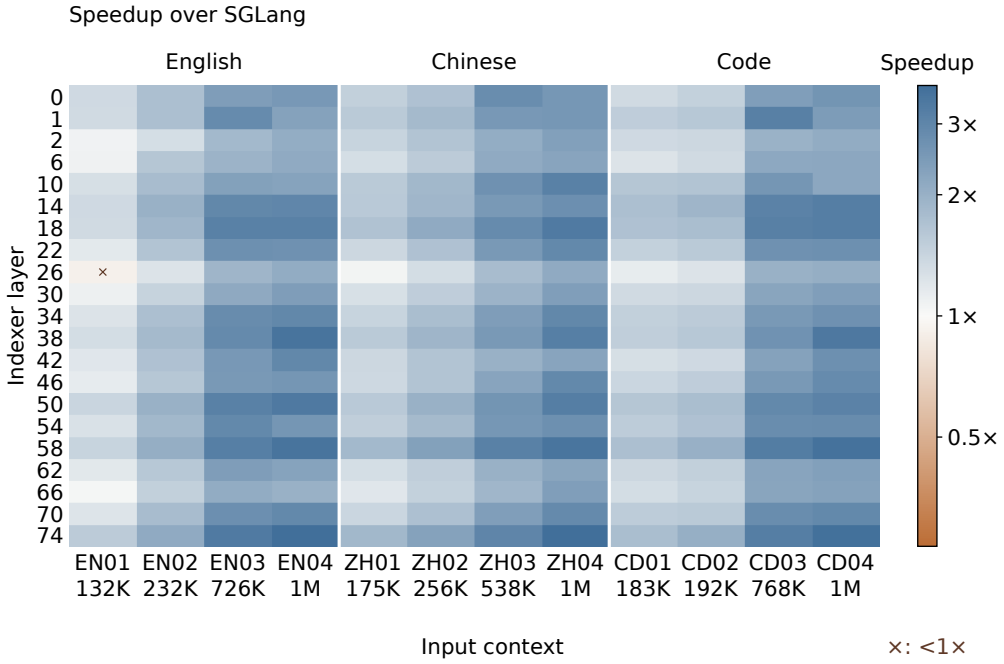


Figure 11: Per-layer retrieval speedup over SGLang for the final 32K queries. Columns identify inputs and context lengths; rows identify indexer layers. Crosses mark cells slower than the baseline. The logarithmic color scale is centered at $1\times$. Five cells use the capacity-repair remeasurements specified in Appendix D.4.

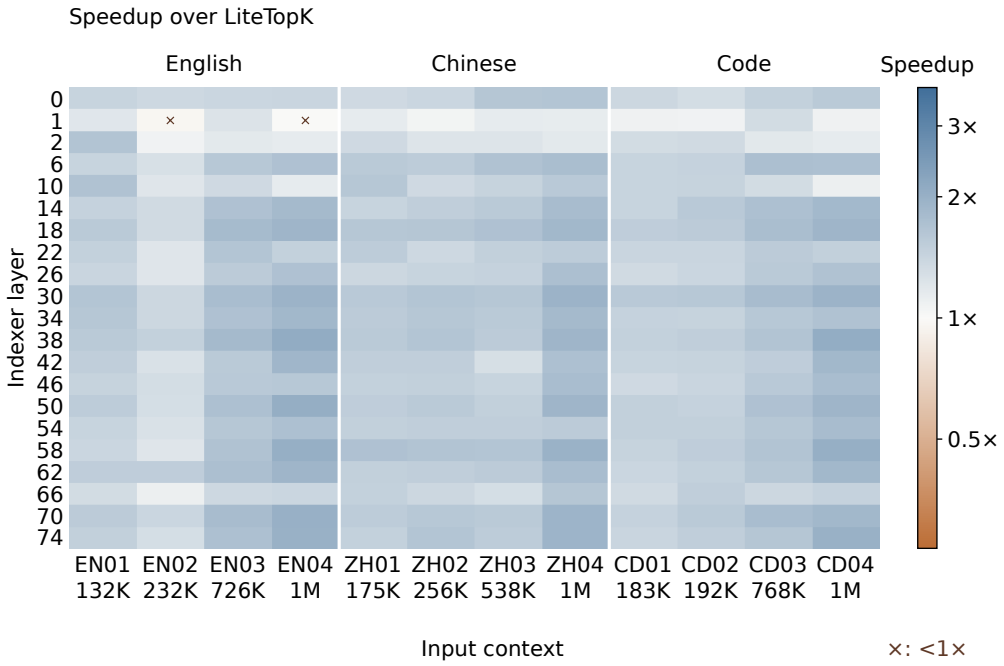


Figure 12: Per-layer retrieval speedup over LiteTopK. Input and layer ordering, timing scope, remeasured cells, and color scale match Figure 11; crosses retain the two comparisons slower than the baseline.