ELSAA: Efficient Low-Rank and Sparse Attention Approximation for Training Transformers

Mahdi Heidari [1] **Mohammad Mahdi Rahimi** [2] **Jaekyun Moon** [1]

Abstract

The quadratic $N \times N$ attention score matrix remains a central obstacle to extending Transformers to longer input lengths. Existing efficient attention methods usually reduce this bottleneck by either imposing sparsity, so that each query attends to only a small subset of keys, or by using low-rank/kernel sketches, so that global interactions are compressed into a lower-dimensional representation. We propose *ELSAA*, an efficient low-rank and sparse approximation of attention. Importantly, ELSAA does *not* decompose the learned projection or output matrices of the Transformer into sparse and low-rank factors. Instead, after dense projections produce Q, K, V, ELSAA approximates the induced attention score operator itself: a sparse branch captures selected high-similarity interactions, while a low-rank branch summarizes diffuse global interactions. Since the two branches can be normalized over supports with very different denominator mass, ELSAA introduces a denominator-aware fusion term that scales the sparse branch according to its estimated attention mass relative to the low-rank branch. This gives a practical framework for constructing low-rank and sparse attention outputs without materializing the full quadratic score matrix, aiming to enable longer-context training while preserving both sharp token-level interactions and broad contextual mixing.

1. Introduction

Transformers have become the dominant architecture for language, vision, and multimodal modeling, largely because self-attention can adaptively mix information across all pairs of tokens (Vaswani et al., 2017; Devlin et al., 2019; Brown et al., 2020; Dosovitskiy et al., 2021; Touvron et al., 2023). However, this pairwise interaction is also the main barrier to extending models to longer input lengths. For a sequence of length N, standard attention forms the score matrix

$$Z = QK^\top/\sqrt{d_h}, \qquad P = \mathrm{softmax}(Z), \tag{1}$$

where $P \in \mathbb{R}^{N\times N}$ is applied to the value matrix V. Equivalently, for a single query token i, dense attention computes

$$d_i = \sum_{j=1}^{N} \exp(z_{ij}), \qquad o_i = \frac{1}{d_i}\sum_{j=1}^{N} \exp(z_{ij}) v_j. \tag{2}$$

Here d_i is the attention denominator, and it determines the scale on which the weighted value average is normalized. The memory and compute cost of this dense $N \times N$ operator grows quadratically with context length. Systems advances such as FlashAttention improve the IO efficiency and practical runtime of exact dense attention, but they do not change the fact that exact attention still represents all pairwise token interactions (Dao et al., 2022; Dao, 2023). Therefore, algorithmic approximations of the attention operator remain essential for training and serving models with substantially longer contexts.

A large body of work reduces the cost of attention by exploiting either *sparsity* or *low-rank structure*. Sparse attention methods restrict each query to a subset of keys, using local windows, global tokens, random patterns, routing, clustering, hashing, or large-entry detection (Child et al., 2019; Beltagy et al., 2020; Zaheer et al., 2020; Kitaev et al., 2020; Roy et al., 2021; Daras et al., 2020; Han et al., 2024b; Jiang et al., 2024). These methods are especially appealing when the attention distribution is low-entropy, peaked, or dominated by a small number of relevant keys. In such regimes, most of the denominator in Equation (2) may be contributed by only a small fraction of the full key set, and computing the corresponding high-mass interactions exactly can preserve the sharp token-level behavior of dense attention. In contrast, low-rank, kernel, and sketch-based methods approximate the dense attention operator through compressed features or landmarks, enabling global mixing without explicitly constructing the full matrix (Wang et al.,

[1]School of Electrical Engineering, KAIST, Daejeon 34141, Republic of Korea [2]Department of Electrical Engineering and Computer Science, Daegu Gyeongbuk Institute of Science and Technology (DGIST), Daegu 42988, Republic of Korea. Correspondence to: Mahdi Heidari <mahdi.heidari.ee.sut@kaist.ac.kr>.

Proceedings of the 43rd International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026. Copyright 2026 by the author(s).

2020; Katharopoulos et al., 2020; Choromanski et al., 2021; Peng et al., 2021; Xiong et al., 2021). These methods are attractive when attention is high-entropy or diffuse, so that information is spread over many tokens and global context is better represented by a compressed dense summary. However, either assumption alone can be brittle: sparse methods may miss weak but globally important interactions, while low-rank methods may smooth out sharp, high-mass token-token dependencies.

This complementarity suggests that attention should often be approximated as a combination of sparse and low-rank components. Prior work has provided strong evidence for this view, showing that the two approximation families can excel in different regimes of the attention matrix (Chen et al., 2021). Classical robust PCA also motivates the broader idea that many matrices can be represented as the sum of a low-rank term and a sparse residual (Candès et al., 2011; Chandrasekaran et al., 2011). Nevertheless, combining these structures inside attention is subtle. A naive addition of sparse attention and low-rank attention can double-count selected interactions, and Scatterbrain's entrywise correction handles this issue at the level of an unnormalized attention estimator by correcting selected entries exactly (Chen et al., 2021). However, an entrywise correction alone does not answer how two separately normalized branches should be balanced during training. In particular, a sparse branch and a low-rank branch may have very different denominator masses, so their outputs can live on different scales even when both branches are useful.

We propose **ELSAA**: **E**fficient **L**ow-rank and **S**parse **A**pproximation of **A**ttention. ELSAA is built around a simple principle: use a sparse branch for high-confidence interactions and a low-rank branch for global residual context. The sparse branch selects keys that are likely to be highly relevant to a given query and computes attention over the selected support. Different sparse key-finding mechanisms can instantiate this principle, including locality-sensitive hashing, routing or clustering, learned or dynamic sparse patterns, and approximate large-entry detection (Kitaev et al., 2020; Roy et al., 2021; Daras et al., 2020; Han et al., 2024b; Jiang et al., 2024; Tang et al., 2024). The low-rank branch summarizes global key/value information in a compressed representation and produces a query-dependent global context vector without materializing all pairwise scores.

For a given query token i, ELSAA forms a fused output

$$O_i = g_{\mathrm{sparse},i}\, m_{\mathrm{sparse},i}\, O_{\mathrm{sparse},i} + g_{\mathrm{lr},i}\, O_{\mathrm{lr},i}. \tag{3}$$

where o_i^{sparse} is the sparse-branch output, o_i^{lowrank} is the low-rank-branch output, and g_{sparse} and g_{lowrank} are learned gates. Let d_i^{sparse} denote the estimated denominator mass covered by the sparse branch, and let d_i^{lowrank} denote the denominator estimate associated with the low-rank branch. We introduce the denominator-aware correction

$$m_{\mathrm{sparse},i} = \frac{d_i^{\mathrm{sparse}}}{d_i^{\mathrm{sparse}} + \lambda d_i^{\mathrm{lowrank}} + \varepsilon}, \tag{4}$$

where λ can be fixed or learned. Several empirical studies of long-context attention and KV-cache compression observe that a small fraction of critical tokens or sparse attention patterns can dominate the attention outcome (Zhang et al., 2023; Xiao et al., 2024; Tang et al., 2024; Jiang et al., 2024). However, directly adding a sparse output to a low-rank output can distort the scale of the true attention computation, especially when the two branches are normalized by denominators of different magnitudes. The role of $m_{\mathrm{sparse},i}$ is to rescale the sparse contribution according to its estimated denominator mass, so that the two outputs are fused on a more comparable scale. Thus, ELSAA does not merely add two attention approximations; it explicitly accounts for their relative normalization scales.

We emphasize a distinction from sparse plus low-rank *weight* parameterizations. Recent work has shown that representing learned matrices using both low-rank and sparse components can improve parameter and memory efficiency for compression, fine-tuning, or pretraining (Li et al., 2023; Han et al., 2024a; Mozaffari et al., 2024). ELSAA addresses a different object. We keep the learned Transformer projections dense and instead approximate the induced $N \times N$ attention operator after Q, K, V have been formed. This distinction is important for long-context modeling: the bottleneck we target is not the number of parameters in W_Q, W_K, W_V, W_O, but the quadratic attention matrix that appears for every input sequence. Combining these two orthogonal directions is also an interesting future research path: one may approximate the learned projection and output matrices in parameter space while simultaneously approximating the input-dependent attention matrix in sequence space.

Our contributions are as follows. First, we formulate long-context attention as a low-rank and sparse operator construction rather than as a decomposition of learned projection matrices. Second, we combine an LSH-selected exact sparse branch with a RACE-style low-rank sketch branch, targeting both sharp local/similar-token interactions and diffuse global context. Third, we introduce denominator-aware fusion through m_{exact}, with optional learnable λ, to address scale mismatch between sparse and sketch attention branches during training.

2. Related Work

Efficient attention for long sequences. Efficient Transformers aim to overcome the quadratic cost of dense self-attention (Tay et al., 2020; 2022). One line of work keeps exact or nearly exact attention but improves practical execu-

tion. FlashAttention uses IO-aware tiling to avoid materializing the full attention matrix in high-bandwidth memory, providing major speed and memory improvements for exact attention (Dao et al., 2022; Dao, 2023). This is complementary to ELSAA: FlashAttention optimizes the computation of dense or block attention, while ELSAA changes the attention operator to avoid constructing all pairwise interactions. Another line of work imposes sparse attention patterns. Sparse Transformer uses structured sparse patterns (Child et al., 2019); Longformer and BigBird combine local, global, and random attention to support long documents (Beltagy et al., 2020; Zaheer et al., 2020); Reformer uses LSH to reduce attention cost (Kitaev et al., 2020); Routing Transformer and SMYRF use clustering or asymmetric hashing to identify relevant key blocks (Roy et al., 2021; Daras et al., 2020); and HyperAttention uses LSH-style detection of large entries to obtain near-linear long-context attention under appropriate structure (Han et al., 2024b). More recent long-context inference methods also exploit dynamic sparse attention or query-aware critical-token selection, further supporting the view that only part of the full attention matrix may be necessary for many inputs (Zhang et al., 2023; Xiao et al., 2024; Tang et al., 2024; Jiang et al., 2024). These methods are effective when attention is sparse or structured, but sparse selection alone may not capture diffuse background interactions across the entire sequence.

Low-rank, kernel, and sketch attention. A second family of efficient attention methods approximates the dense attention operator with low-dimensional features. Linformer projects the sequence dimension and assumes that attention matrices are approximately low-rank (Wang et al., 2020). Linear Transformer rewrites attention using kernel feature maps so that computation can be associated as $\phi(Q)(\phi(K)^\top V)$ (Katharopoulos et al., 2020). Performer uses positive random features to approximate softmax attention with linear complexity (Choromanski et al., 2021), while Random Feature Attention studies related random-feature approximations for autoregressive modeling (Peng et al., 2021). Nyströmformer approximates attention using landmark-based Nyström methods (Xiong et al., 2021). RACE attention uses differentiable hash-sketch summaries, where keys are softly assigned to buckets, bucket statistics are accumulated, and queries read out from these summaries (Joshi et al., 2026). Such methods provide global context at subquadratic or linear cost, but they can struggle to represent sharp, high-confidence entries of the attention distribution. ELSAA uses the low-rank branch for global coverage while delegating high-mass interactions to a sparse branch.

Sparse plus low-rank attention. The idea that attention may require both sparse and low-rank structure is closely related to sparse plus low-rank matrix decomposition. Robust PCA studies decompositions of a matrix into a low-rank component and a sparse component (Candès et al., 2011; Chandrasekaran et al., 2011). Scatterbrain applies this perspective to attention and shows that sparse and low-rank approximations can be complementary across attention entropy regimes (Chen et al., 2021). Its construction uses a low-rank random feature approximation together with a sparse LSH-selected correction so that selected entries match the unnormalized exponential attention exactly, while non-selected entries are approximated by the low-rank component. This provides an elegant entrywise estimator with unbiasedness and reduced variance. However, its analysis is primarily entrywise and does not directly provide a general output-level variance or scale analysis for combining attention branches that may be normalized by different denominators. ELSAA takes a related but distinct view: rather than correcting a single unnormalized matrix estimator, we study a training-time architecture with two separately normalized attention-like branches. In this setting, the main issue is not only entrywise unbiasedness, but also how to balance branch outputs whose denominator masses and scales can differ. The factor $m_{\mathrm{sparse},i}$ in Equation (4) is designed for this branch-scale mismatch.

Sparse plus low-rank parameterization of model weights. Sparse plus low-rank structure has also been used to parameterize, compress, or adapt the learned weights of neural networks. LoRA represents fine-tuning updates with low-rank factors (Hu et al., 2022), while subsequent work studies quantized low-rank adaptation, weight-decomposed low-rank adaptation, restarted low-rank training, and low-rank gradient projection (Dettmers et al., 2023; Liu et al., 2024; Lialin et al., 2024; Zhao et al., 2024). Other work explicitly combines sparsity with low-rank structure in parameter space: LoSparse compresses model weights using low-rank and sparse approximations (Li et al., 2023); SLoPe augments sparse LLM pretraining with low-rank adapters (Mozaffari et al., 2024); and SLTrain parameterizes weight matrices with a learned low-rank component together with a learned sparse component, improving parameter and memory efficiency during pretraining (Han et al., 2024a). Sparse low-rank adaptation methods also study how sparsity can be incorporated into low-rank fine-tuning updates (Ding et al., 2023). This direction is orthogonal to ELSAA. We do not replace W_Q, W_K, W_V, or W_O by sparse plus low-rank factors. Instead, our goal is to approximate the input-dependent attention operator induced by dense projections, thereby reducing the sequence-length bottleneck and enabling longer contexts. In short, SL-style methods decompose learned matrices in parameter space, whereas ELSAA constructs a sparse plus low-rank approximation in attention space.

Table 1. Comparison of attention methods across long-context classification tasks.

Method	ArXiv @ 32K		Oxford-IIIT Pet @ 16K		Flowers-102 @ 16K		Food-101 @ 16K		Average	
	Train ↑	Test ↑	Train ↑	Test ↑	Train ↑	Test ↑	Train ↑	Test ↑	Train ↑	Test ↑
ELSAA	99.97%	93.93%	**91.55%**	**22.51%**	**97.84%**	42.45%	**76.43%**	**28.33%**	**91.45%**	**46.81%**
Sort_LSH	95.41%	84.94%	23.96%	12.65%	77.25%	37.25%	59.63%	19.03%	64.06%	38.47%
RACE	**100.00%**	**95.05%**	59.89%	16.13%	91.66%	35.49%	67.57%	24.00%	79.78%	42.67%
Sort_Lsh_RACE	99.95%	93.60%	76.76%	19.00%	95.98%	**42.65%**	76.05%	26.67%	87.19%	45.48%
Exactflash	81.77%	70.11%	92.60%	24.66%	97.81%	48.13%	78.93%	29.96%	87.78%	43.22%

Table 2. Text Retrieval @ 64K

Method	Train Acc. ↑	Test Acc. ↑
ELSAA	94.73%	65.34%
RACE	**94.95%**	**66.30%**
Sort_Lsh_RACE	94.28%	66.00%
Exactflash	≈ 50	≈ 50

3. Methodology

We instantiate ELSAA inside each attention layer by replacing dense attention with a sparse–low-rank hybrid operator that avoids constructing the full $N \times N$ score matrix. In this work, we use RACE attention as the low-rank global branch (Joshi et al., 2026), and use sorted Hamming LSH to identify highly correlated query-key pairs with high probability. This follows a line of efficient-attention methods that use hashing, kernel-density estimation, or large-entry detection to avoid materializing all pairwise attention scores, including KDEformer and HyperAttention (Zandieh et al., 2023; Han et al., 2024b). The sparse branch captures high-similarity query-key interactions exactly, while the low-rank branch supplies global context through compressed bucket statistics. Future variants should also consider sliding-window, global-token, and learned top-k sparse selectors; these are natural alternatives but require careful implementation to preserve efficiency.

We describe the method for one attention head. Multi-head ELSAA applies the same procedure independently per head and concatenates the outputs as in standard Transformers. Let $Q, K, V \in \mathbb{R}^{N \times d_h}$ be the query, key, and value matrices. Dense attention computes $O_i = \frac{\sum_{j=1}^{N} \exp(Q_i^\top K_j/\sqrt{d_h})V_j}{\sum_{j=1}^{N} \exp(Q_i^\top K_j/\sqrt{d_h})}$, which requires all N^2 query-key scores. ELSAA instead computes two outputs, O_{lr} and O_{sparse}, together with denominator proxies $d_{\mathrm{lr}}, d_{\mathrm{sparse}}$, and combines them through a denominator-aware fusion rule.

3.1. Low-rank branch: RACE attention

RACE replaces the dense attention matrix with soft hash-bucket summaries. Each table softly assigns queries and keys to hypercube buckets, accumulates key/value statistics inside these buckets, and lets each query read from the resulting summaries. For a fixed number of tables, buckets, and head dimension, this gives a linear-time global branch in the sequence length N. We use the normalized ratio form because it directly provides both a low-rank output O_{lr} and a denominator proxy d_{lr}, which are both needed by the ELSAA fusion rule. For the causal setting, we extend RACE by partitioning the sequence into fixed-size chunks and decomposing the output into an inter-chunk contribution—bucket statistics accumulated from all strictly prior chunks via prefix sums—and a causally-masked intra-chunk contribution, preserving linear complexity. Full pseudocode for the non-causal and causal variants is given in Section A and Algorithms 2 and 3.

3.2. Sparse branch: sortLSH exact attention

The sparse branch is designed to preserve sharp token-level interactions that the low-rank branch may smooth. We use a sortLSH selector: queries and keys are hashed, sorted by their hash values, and grouped into equal-size blocks. After sorting, high-similarity query-key interactions are expected to concentrate near diagonal blocks of the permuted attention matrix. This lets us compute exact attention inside selected blocks using dense block matrix multiplications, without constructing the full $N \times N$ attention matrix. The branch returns a normalized sparse output O_{sparse} and its denominator d_{sparse}. For the causal setting, we adopt a recursive divide-and-conquer structure (Han et al., 2024b): the sequence is split into past and future halves, the diagonal blocks are processed by recursive causal calls, and the future–past off-diagonal block is handled by standard non-causal sortLSH since all its keys precede all its queries; the two future contributions are merged via a log-sum-exp-aware combination that preserves exact softmax normalization. Full pseudocode for the non-causal and causal variants is given in Section A and Algorithms 4 and 5.

3.3. Denominator-aware sparse–low-rank fusion

The two branches are complementary but live on different normalization scales. The sparse branch computes selected high-confidence interactions exactly, but its support may cover only part of the row denominator. The low-rank branch provides global coverage, but may smooth sharp interactions. Therefore, directly adding the two outputs can

Table 3. Comparison of attention methods on IMDB and Fashion-MNIST.

Method	IMDB @ 512		Fashion-MNIST @ 784		Average	
	Train ↑	Test ↑	Train ↑	Test ↑	Train ↑	Test ↑
ELSAA	**90.68%**	75.93%	87.79%	88.25%	89.24%	82.09%
Sort_LSH	70.45%	66.14%	82.11%	83.15%	76.28%	74.65%
RACE	89.93%	77.41%	**89.79%**	**89.43%**	**89.86%**	**83.42%**
Sort_Lsh_RACE	90.20%	**77.71%**	85.43%	86.39%	87.82%	82.05%
Exactflash	89.54%	78.67%	85.95%	85.81%	87.75%	82.24%

Table 4. NIAH task test accuracy across input lengths. OOM indicates out-of-memory.

Method	512	1024	2048	4096	8192	16384	32768	65536	Avg.
Exact Flash	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	OOM	OOM	100.0%
ELSAA	**100.0%**	**100.0%**	**100.0%**	**100.0%**	**100.0%**	**100.0%**	98.8%	84.2%	**97.88%**
Sparse_LSH	24.6%	**100.0%**	**100.0%**	**100.0%**	**100.0%**	**100.0%**	**100.0%**	**100.0%**	90.58%
RACE	11.0%	**100.0%**	**100.0%**	52.0%	13.4%	6.4%	4.8%	2.2%	36.23%
Performer	19.2%	**100.0%**	88.2%	87.6%	88.2%	87.4%	83.2%	83.4%	79.65%

Table 5. Comparison of **causal** attention methods across long-context classification tasks.

Method	ArXiv @ 32K		ArXiv @ 64K		Tiny ImageNet @ 1024		Average	
	Train ↑	Test ↑	Train ↑	Test ↑	Train ↑	Test ↑	Train ↑	Test ↑
ELSAA	**100%**	**88.90%**	**100%**	**91.47%**	**94.44%**	36.07%	**98.15%**	**72.15%**
RACE	99.97%	87.84%	99.88%	91.46%	93.41%	**36.39%**	97.75%	71.90%
Exactflash	99.97%	87.96%	99.83%	87.44%	96.59%	37.32%	98.80%	70.91%

over-amplify the sparse branch or double count overlapping mass.

ELSAA introduces a denominator-aware sparse multiplier

$$m_{\text{sparse},i} = \frac{d_{\text{sparse},i}}{d_{\text{sparse},i} + \lambda_i d_{\text{lr},i} + \varepsilon}, \tag{5}$$

where $\lambda_i > 0$ can be fixed, a learnable scalar shared across tokens, or a query-dependent coefficient. The final ELSAA output uses two learned gates:

$$(g_{\text{sparse},i}, g_{\text{lr},i}) = \sigma(G_\theta(q_i)) \in (0,1)^2.$$

The fused output for token i is

$$O_i = g_{\text{sparse},i}\, m_{\text{sparse},i}\, O_{\text{sparse},i} + g_{\text{lr},i}\, O_{\text{lr},i}. \tag{6}$$

The factor $m_{\text{sparse},i}$ scales the sparse branch according to its estimated denominator mass relative to the low-rank branch, while the gates learn how much each branch should contribute to the final representation.

Practical notes. The RACE branch can be implemented with different reduction modes using the same bucket statistics, while the sparse branch can be replaced by any efficient important-key selector that returns an exact sparse output and a denominator. In this work, we use sortLSH blocks because sorting makes the selected attention matrix block-diagonal after permutation, enabling efficient exact block attention without materializing the full $N \times N$ matrix.

4. A Rank View of Sparse + Low-Rank Attention

We study hybrid sparse–low-rank attention operators of the form $\widehat{M} = S_\Omega + BA$, where $S_\Omega \in \mathbb{R}^{n\times n}$ is sparse and $BA \in \mathbb{R}^{n\times n}$ has rank at most r, with $B \in \mathbb{R}^{n\times r}$, $A \in \mathbb{R}^{r\times n}$. The sparse support $\Omega \subseteq [n] \times [n]$ is generated by angular collisions.

Definition 4.1 (Angular collision probability)**.** For nonzero $Q_i, K_j \in \mathbb{R}^d$, define

$$\rho_{ij} := 1 - \frac{1}{\pi} \arccos \frac{Q_i^\top K_j}{\|Q_i\|_2 \|K_j\|_2} \in [0,1].$$

For $\gamma > 0$, the single-trial collision probability is $\pi_{ij} := \rho_{ij}^\gamma$. With L_s independent sparse hash trials, including (i,j) after at least one collision gives

$$q_{ij} := 1 - (1 - \pi_{ij})^{L_\text{s}} = 1 - (1 - \rho_{ij}^\gamma)^{L_\text{s}}.$$

Let $M^\star \in \mathbb{R}^{n\times n}$ denote the exact attention score/kernel matrix, e.g., $M^\star_{ij} = \exp(Q_i^\top K_j)$, and define $S_\Omega := \Omega \odot M^\star$.

Definition 4.2 (Sparse collision graph and matching deficiency)**.** The support Ω defines the bipartite graph $G_\Omega = ([n]_\text{row}, [n]_\text{col}, \Omega)$, where edge $(i,j) \in \Omega$ connects row i to column j. Let $\nu(\Omega)$ be its maximum matching size and define the matching deficiency $d(\Omega) := n - \nu(\Omega)$.

Algorithm 1 ELSAA: Efficient Low-Rank and Sparse Approximation of Attention

1: **Input:** hidden states $X \in \mathbb{R}^{N\times d}$, projections W_Q, W_K, W_V, W_O, RACE parameters $L_{\mathrm{s}}, \gamma, \beta$, sortLSH block size b, gate network G_θ, coefficient rule $\lambda_i > 0$, numerical floor $\varepsilon > 0$
2: **Output:** hybrid attention output $O \in \mathbb{R}^{N\times d}$
3: Project to one attention head

$$Q \leftarrow XW_Q, \qquad K \leftarrow XW_K, \qquad V \leftarrow XW_V.$$

4: Run the low-rank branch

$$(O_{\mathrm{lr}}, d_{\mathrm{lr}}) \leftarrow \mathrm{RACE}(Q, K, V; L_{\mathrm{s}}, \gamma, \beta, \varepsilon).$$

5: Run the sparse exact branch

$$(O_{\mathrm{sparse}}, d_{\mathrm{sparse}}) \leftarrow \mathrm{sortLSH}(Q, K, V; b, \varepsilon).$$

6: Compute denominator-aware sparse multiplier

$$m_{\mathrm{sparse},i} \leftarrow \frac{d_{\mathrm{sparse},i}}{d_{\mathrm{sparse},i} + \lambda_i d_{\mathrm{lr},i} + \varepsilon}, \qquad i = 1, \dots, N.$$

7: Compute token-wise gates

$$(g_{\mathrm{sparse}}, g_{\mathrm{lr}}) \leftarrow \sigma(G_\theta(Q)).$$

8: Fuse branch outputs

$$O_{\mathrm{head}} \leftarrow g_{\mathrm{sparse}} \odot m_{\mathrm{sparse}} \odot O_{\mathrm{sparse}} + g_{\mathrm{lr}} \odot O_{\mathrm{lr}}.$$

9: Return $O \leftarrow O_{\mathrm{head}} W_O$

Assumption 4.3 (Independent angular edge model)**.** Conditioned on Q, K, we analyze the idealized model $\Omega_{ij} \sim \mathrm{Bernoulli}(q_{ij})$ independently over i, j, preserving the marginal angular collision probabilities.

Assumption 4.4 (Generic sparse values and generic low-rank factors)**.** The nonzero entries of S_Ω are in general position on their support, and B, A are drawn independently from absolutely continuous distributions.

Proposition 4.5 (Sparse matching deficiency plus low rank)**.** *Under Theorem 4.4, for fixed Ω,*

$$\mathrm{rank}(S_\Omega + BA) = \min\{n, \nu(\Omega) + r\} \quad \text{a.s.}$$

Hence, if $\nu(\Omega) \geq n - r$, then $\mathrm{rank}(S_\Omega + BA) = n$ almost surely.

Thus, the sparse component contributes rank through the maximum matching of its support graph, while the rank-r branch fills up to r missing directions.

Theorem 4.6 (Full rank from angular sparse collisions and rank-r low rank)**.** *Condition on Q, K, and suppose Theorems 4.3 and 4.4 hold. Let $R := r + 1$, and for $I, U \subseteq [n]$ define*

$$\Lambda(I, U) := \sum_{i\in I,\ j\in U} q_{ij},$$

$$\Delta_r(Q, K) := \sum_{\substack{I,U\subseteq[n] \\ |I|\geq R,\ |U|\geq R,\ |I|+|U|\geq n+R}} e^{-\Lambda(I,U)}.$$

Then

$$\mathbb{P}[\mathrm{rank}(S_\Omega + BA) = n \,|\, Q, K] \geq 1 - \Delta_r(Q, K).$$

The sets I, U in $\Delta_r(Q, K)$ are Hall-deficiency cuts: failure occurs when a large query set has no sparse edges into a large key set. The quantity $\Lambda(I, U)$ is the expected number of sampled sparse entries in that rectangle, so full rank holds when every Hall-relevant rectangle has enough expected angular collision mass.

We prove the theoretical claims from this section in Section B. In Section C, we further analyze settings that simplify Theorem 4.6 to $\mathbb{P}[\mathrm{rank}(S_\Omega + BA) = n \,|\, Q, K] \geq 1 - n^{-c}$. These results show that $S_\Omega + BA$ can be full rank with high probability under natural sparse-coverage conditions, motivating further study of sparse–low-rank attention rank behavior.

5. Experimental Setup

We evaluate ELSAA across classification and retrieval tasks to test whether the sparse–low-rank attention structure is useful beyond a single modality or dataset. Since our main motivation is efficient long-context modeling, we prioritize settings with relatively long input sequences or large tokenized inputs whenever possible. The benchmark suite includes long-document text classification, sentiment classification, fine-grained image classification, low-resolution image classification, and text retrieval. For all tasks, we compare attention variants under the same training and evaluation settings, and we report top-1 accuracy. Full task-specific hyperparameters, including learning rates, batch sizes, sequence lengths, model sizes, and training schedules, are reported in Section D.

Datasets and tasks. We evaluate on text, image, and retrieval benchmarks: ArXiv for scientific text classification, IMDB for binary sentiment classification (Maas et al., 2011), Food-101 (Bossard et al., 2014), Fashion-MNIST (Xiao et al., 2017), Flowers-102 (Nilsback & Zisserman, 2008), and Oxford-IIIT Pet (Parkhi et al., 2012). For ArXiv classification, the model predicts the document subject class; for text retrieval, two ArXiv documents are paired into a 64K-token sequence and the model performs binary same-class prediction. We also evaluate on a synthetic Needle-in-a-Haystack (NIAH) benchmark, where a key-value needle is inserted at a random position in a long distractor context and the model must recall the value at sequence end. NIAH probes content-based retrieval across lengths

$\{512, 1024, 2048, 4096, 8192, 16384, 32768, 65536\}$, with a separate model trained on length 1024. Finally, we evaluate the causal variants of Algorithm 6 on autoregressive long-context classification: packed ArXiv at $N \in \{32{,}000, 64{,}000\}$, where the classification head reads the last valid token under strict left-to-right access, and Tiny ImageNet (200 classes, 64×64) using a causal ViT with patch size 2 ($N = 1024$ tokens, raster-order causal mask, class token at position 0). Together these tasks span long text, tokenized vision, retrieval, and causal aggregation across a wide range of sequence lengths.

Attention variants. We compare **Exactflash** as the full-attention baseline, **RACE** as the low-rank baseline, **Sort_Lsh** as the sparse baseline, and **ELSAA** as our proposed sparse–low-rank method. We also include **Sort_Lsh_RACE** as an ablation that combines both branches but sets $m_{\text{sparse}} = 1$, removing denominator-aware rescaling. For the causal experiments, Sort_Lsh and Sort_Lsh_RACE are omitted, as their sparse branch is replaced by the recursive causal sortLSH of Algorithm 5—which is precisely the sparse branch inside causal ELSAA—making them non-comparable baselines. In all result tables, we bold the highest accuracy among efficient attention variants, excluding Exactflash as it does not reduce the quadratic computation cost.

Hardware and evaluation protocol. All experiments were run on an NVIDIA RTX PRO 6000 Blackwell GPU with 48 GB of memory. Within each task, all attention variants use the same model, training budget, optimizer, and evaluation protocol. Unless otherwise stated, we report top-1 accuracy.

6. Complexity of ELSAA

We analyze the attention-mixing cost for a non-causal sequence of length N, suppressing constants from batch size and number of heads. Exactflash attention computes all query–key interactions:

$$\mathcal{C}_{\text{Exactflash}} = \Theta(N^2), \qquad \mathcal{M}_{\text{Exactflash}} = \Theta(N^2).$$

In **Sort_Lsh**, tokens are sorted by angular LSH and exact attention is computed only within fixed-size sorted blocks of size s. Therefore each query attends to at most s keys:

$$\mathcal{C}_{\text{Sort_Lsh}} = \Theta(Ns), \qquad \mathcal{M}_{\text{Sort_Lsh}} = \Theta(Ns).$$

With r neighboring sorted blocks, this becomes $\Theta(N(2r+1)s)$. In our main setting $r = 0$, so the branch cost is $\Theta(Ns)$.

For **Race**, let L_s be the number of hash tables and γ be the number of hash bits per table. Each table has 2^γ buckets, hence the total number of bucket features is

$$S_R = L_s 2^\gamma.$$

Race does not materialize an $N \times N$ matrix; it builds global bucket summaries and each query reads from these S_R features:

$$\mathcal{C}_{\text{Race}} = \Theta(NL_s 2^\gamma), \qquad \mathcal{M}_{\text{Race}} = \Theta(NL_s 2^\gamma).$$

ELSAA combines Sort_Lsh and Race, with denominator-aware sparse rescaling:

$$\mathcal{C}_{\text{ELSAA}} = \Theta\big(Ns + NL_s 2^\gamma\big) = \Theta\big(N(s + L_s 2^\gamma)\big),$$

$$\mathcal{M}_{\text{ELSAA}} = \Theta\big(N(s + L_s 2^\gamma)\big).$$

The ablation **Sort_Lsh_RACE** has the same asymptotic cost as ELSAA, but removes the rescaling term by setting $m_{\text{sparse}} = 1$.

For fixed s, L_s, and γ, Sort_Lsh, Race, Sort_Lsh_RACE, and ELSAA are linear in N, while Exactflash attention is quadratic.

Complexity summary. ELSAA reduces the quadratic N^2 attention interaction cost by combining a sparse branch with $O(Ns)$ selected query–key interactions and a low-rank RACE branch with $O(NL_{\text{s}}2^\gamma)$ bucket interactions, where s is the sparse block budget, L_{s} is the number of hash tables, and γ is the number of hash bits. Thus, ignoring lower-order hashing and sorting overhead, the attention-interaction cost scales as

$$O(N(s + L_{\text{s}}2^\gamma))$$

instead of $O(N^2)$. We provide a concrete numerical example in Section E, showing that ELSAA reduces attention interactions by approximately 99%.

7. Results and Discussion

Tables 5–4 compare ELSAA with sparse, low-rank, hybrid, and exact full-attention baselines across long-context vision, long-context text, retrieval, short-sequence, and length-extrapolation settings. We organize the discussion around four observations: (i) ELSAA improves on Sort_Lsh_RACE consistently, isolating the contribution of the denominator-aware correction; (ii) the relative strength of the sparse and low-rank branches reflects a fundamental structural difference between vision and long-text tasks; (iii) exact full attention is not merely expensive at very long contexts but can fail to optimize at all; and (iv) ELSAA is the only method that extrapolates reliably across all lengths on NIAH, including lengths shorter than training.

Isolating the denominator-aware correction. One important ablation compares Sort_Lsh_RACE with ELSAA. These two methods share identical architecture, branches, gates, and hyperparameters; the only difference is that Sort_Lsh_RACE sets $m_{\text{sparse}} = 1$, while ELSAA computes the denominator-aware multiplier of Equation (5). Any performance gap is therefore attributable solely to this single term. On the long-context benchmarks of Table 5, ELSAA achieves an average test accuracy of 46.81% versus 45.48% for Sort_Lsh_RACE, a consistent $+1.33$pp improvement obtained without changing parameter count, branch complexity, or optimization budget. The improvement is most visible on structured vision tasks (Oxford-IIIT Pet: 22.51% vs. 19.00%; Food-101: 28.33% vs. 26.67%), supporting our central claim: when two branches are normalized over supports of very different denominator mass, naive addition distorts the fused output, and m_{sparse} corrects this with a principled rescaling rather than a learned heuristic.

Vision is structured; long-text classification is diffuse. The most striking pattern in Table 5 is that the relative ranking of RACE, Sort_LSH, and ELSAA changes systematically between long text and long vision. On long vision tasks ELSAA dominates, improving over RACE by $+6.4$pp on Oxford-IIIT Pet, $+7.0$pp on Flowers-102, and $+4.3$pp on Food-101. On ArXiv @ 32K, RACE alone (95.05%) marginally outperforms ELSAA (93.93%). This dichotomy is structural: vision attention is locally peaked, dominated by a small support of high-mass interactions that an unconditional low-rank approximation smooths away, whereas long-document topic classification requires aggregating diffuse weak evidence across the sequence, making a low-rank summary not only sufficient but preferable. ELSAA handles both regimes within a single architecture, consistent with the rank perspective of Section 4.

Exact attention fails at very long context. The Text Retrieval @ 64K results in Table 2 are decisive. ExactFlash collapses to approximately 50% on both train and test—random performance on a binary task—while ELSAA reaches 65.34% and RACE 66.30%. Dense attention at $N = 64{,}000$ spreads gradient signal across $\approx 4 \times 10^9$ pairwise interactions per layer, the vast majority uninformative. Approximate attention acts as an implicit structural prior that focuses the model on a tractable subset of interactions and stabilizes optimization, enabling a regime that exact attention cannot reach. The broad-aggregation nature of retrieval again favors the low-rank branch slightly, with RACE outperforming ELSAA by ≈ 1pp—consistent with the structural argument above.

Short tasks do not require hybrid structure. Table 3 confirms the natural complement: at $N = 512$ and $N = 784$, ELSAA, RACE, and ExactFlash are all within 1–2pp of each other. The relevant observation is not that ELSAA wins, but that it degrades gracefully rather than failing—an important property for a general-purpose attention layer deployed across heterogeneous workloads.

NIAH: length extrapolation and the role of each branch. Table 4 reports NIAH accuracy when all models are trained at $N = 1024$ and evaluated across lengths $\{512, \ldots, 65536\}$, probing both shorter-than-training and longer-than-training generalization. Several findings stand out. First, RACE fails catastrophically beyond the training length: accuracy collapses from 100% at 1024 to 52% at 4096 and near-random (2.2%) at 65536. This confirms that the low-rank bucket summaries, while globally expressive at seen lengths, cannot locate a specific needle token when the sequence length shifts the effective bucket density. Second, ExactFlash achieves perfect accuracy up to 16384 but runs out of memory at 32768 and beyond, highlighting the wall exact attention hits at extreme lengths. Third, Sparse_LSH extrapolates well at long lengths (100% at 32768 and 65536) because content-based hashing is inherently position-independent, but it fails at $N = 512$—shorter than training—with only 24.6% accuracy, suggesting that its block structure becomes degenerate when the sequence is too short to populate meaningful hash buckets. ELSAA, by contrast, achieves 100% at $N = 512$, outperforming Sparse_LSH at this sub-training length: the low-rank RACE branch provides reliable global context when the sparse branch's bucket structure is under-populated, and the denominator-aware gate learns to up-weight it accordingly. At the long end, ELSAA reaches 98.8% at 32768 and 84.2% at 65536—well above RACE and Performer, and within reach of Sparse_LSH—demonstrating that the hybrid structure preserves the content-based retrieval strength of the sparse branch while preventing the catastrophic failure of the low-rank branch alone.

Causal ELSAA matches or exceeds baselines across lengths and modalities. Table 5 (causal) shows that causal ELSAA achieves the highest average test accuracy (72.15%) across ArXiv @ 32K, ArXiv @ 64K, and Tiny ImageNet @ 1024, outperforming causal RACE (71.90%) and causal ExactFlash (70.91%). On the two long-text settings, all three methods are competitive, consistent with the diffuse-attention argument above. On Tiny ImageNet, causal ExactFlash leads slightly in test accuracy (37.32% vs. 36.07% for ELSAA and 36.39% for RACE), suggesting that at $N = 1024$ the quadratic baseline remains viable and the sparse branch's advantage is modest. Crucially, at ArXiv @ 64K causal ExactFlash lags ELSAA by 4pp (87.44% vs. 91.47%), confirming that the optimization difficulty of dense attention at very long contexts persists in the causal setting.

8. Future Work

Scaling laws. Our experiments fix the model size and sweep over sequence length and modality, but a systematic study of how the relative contributions of the sparse and low-rank branches evolve with model scale, data scale, and context length would clarify when each branch dominates, and would inform the design of branch-specific schedules during pretraining. In particular, the dichotomy we observe between peaked (vision) and diffuse (long text) regimes suggests that the appropriate gate and λ schedules may also depend on depth and head index.

Fused causal GPU kernels. A natural systems-level extension is a fused GPU kernel that combines, in one launch, the recursive sortLSH grouping of Algorithm 5, block-wise exact attention within sorted blocks, the chunked cumulative bucket aggregation of Algorithm 3, and the denominator-aware fusion of Equation (5). Implementations along the lines of FlashAttention show that such fusions are feasible and can recover most of the theoretical advantage in practice. A particularly promising aspect of ELSAA in this respect is that the sparse and low-rank branches are independent given Q, K, V, so they admit parallel scheduling on independent CUDA streams or within a single persistent kernel that shares the Q, K, V loads from high-bandwidth memory.

Pretraining of decoder language models. Having validated the causal variant on long-context classification and on retrieval-style NIAH tasks at lengths up to 64K, the next milestone is autoregressive language-model pretraining with causal ELSAA. The empirical literature on long-context inference (Zhang et al., 2023; Xiao et al., 2024; Tang et al., 2024) consistently documents heavy-tailed attention access patterns dominated by attention sinks, recent context, and a small number of task-relevant pivots, which is precisely the structure that the sparse branch is designed to capture. The denominator-aware multiplier m_{sparse} additionally provides a principled account of the mass discarded by KV-cache eviction methods, which currently address this scale problem with heuristics.

Joint compression in parameter and attention space. ELSAA is orthogonal to weight-space compression methods such as LoRA (Hu et al., 2022), LoSparse (Li et al., 2023), and SLTrain (Han et al., 2024a). Approximating the input-dependent attention operator and the learned projection matrices simultaneously is a natural composition: the former targets the sequence-length bottleneck, while the latter targets the parameter-count bottleneck. We see this composition as a promising route to long-context efficient pretraining of large language models.

Theoretical analysis of branch fusion. The rank analysis of Section 4 establishes an expressivity guarantee for the hybrid sparse–low-rank operator but does not directly analyze the bias and variance of ELSAA's normalized branch outputs under the denominator-aware fusion rule. A complementary output-level bias–variance analysis, in the spirit of Scatterbrain's entrywise analysis but at the level of normalized attention outputs, would clarify when the denominator-aware multiplier provably reduces the mean-squared error of the fused estimator, and when learnable λ is preferable to a fixed schedule.

9. Conclusion

We introduced ELSAA, a sparse–low-rank approximation of attention combining an exact sortLSH sparse branch with a low-rank RACE branch, fused via a denominator-aware multiplier m_{sparse} that corrects the scale mismatch between separately-normalized branches. Our rank analysis shows the sparse component contributes rank through its support graph's matching structure while the low-rank component fills the remaining deficiency. Empirically, ELSAA dominates efficient-attention baselines on long-context structured vision, remains competitive on diffuse long-text tasks, and succeeds where exact attention fails entirely (Text Retrieval @ 64K). On NIAH, ELSAA achieves perfect retrieval up to 16K and remains strong at 32K–64K where exact attention runs out of memory and RACE degrades sharply. We further extended both branches to the causal setting and validated causal ELSAA on autoregressive ArXiv classification up to 64K tokens and on Tiny ImageNet, matching or exceeding causal exact attention and causal RACE throughout. The method is linear in sequence length, adds negligible parameters, and degrades gracefully at short lengths, offering a principled path toward long-context attention that is tractable where exact attention is not.

10. Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grants funded by the Korea government (MSIT) (Nos. RS-2024-00340966 and RS-2024-00408003), and by the Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. RS-2024-00444862).

11. Impact statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Beltagy, I., Peters, M. E., and Cohan, A. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

Bossard, L., Guillaumin, M., and Van Gool, L. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision (ECCV)*, pp. 446–461, 2014.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, 2020.

Candès, E. J., Li, X., Ma, Y., and Wright, J. Robust principal component analysis? *Journal of the ACM*, 58(3):1–37, 2011.

Chandrasekaran, V., Sanghavi, S., Parrilo, P. A., and Willsky, A. S. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21(2):572–596, 2011.

Chen, B., Dao, T., Winsor, E., Song, Z., Rudra, A., and Ré, C. Scatterbrain: Unifying sparse and low-rank attention approximation. In *Advances in Neural Information Processing Systems*, 2021.

Child, R., Gray, S., Radford, A., and Sutskever, I. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

Choromanski, K., Likhosherstov, V., Dohan, D., Song, X., Gane, A., Sarlos, T., Hawkins, P., Davis, J., Mohiuddin, A., Kaiser, L., et al. Rethinking attention with performers. In *International Conference on Learning Representations*, 2021.

Dao, T. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023.

Dao, T., Fu, D. Y., Ermon, S., Rudra, A., and Ré, C. Flashattention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems*, 2022.

Daras, G., Kitaev, N., Odena, A., and Dimakis, A. G. SMYRF: Efficient attention using asymmetric clustering. *arXiv preprint arXiv:2010.05315*, 2020.

Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems*, 2023.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, 2019.

Ding, N., Lv, X., Wang, Q., Chen, Y., Zhou, B., Liu, Z., and Sun, M. Sparse low-rank adaptation of pre-trained language models, 2023.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Han, A., Li, J., Huang, W., Hong, M., Takeda, A., Jawanpuria, P., and Mishra, B. SLTrain: A sparse plus low-rank approach for parameter and memory efficient pretraining. In *Advances in Neural Information Processing Systems*, 2024a.

Han, I., Jayaram, R., Karbasi, A., Mirrokni, V., Woodruff, D. P., and Zandieh, A. Hyperattention: Long-context attention in near-linear time. In *International Conference on Learning Representations*, 2024b.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Jiang, H., Li, Y., Zhang, C., Wu, Q., Luo, X., Ahn, S., Han, Z., Abdi, A. H., Li, D., Lin, C.-Y., Yang, Y., and Qiu, L. MInference 1.0: Accelerating pre-filling for long-context LLMs via dynamic sparse attention. In *Advances in Neural Information Processing Systems*, 2024.

Joshi, S., Chowdhury, A., Kanakamedala, A., Singh, E., Tu, E., and Shrivastava, A. RACE attention: A strictly linear-time attention layer for training on outrageously large contexts. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=RR8Lh8RHgA.

Katharopoulos, A., Vyas, A., Pappas, N., and Fleuret, F. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, 2020.

Kitaev, N., Kaiser, L., and Levskaya, A. Reformer: The efficient transformer. In *International Conference on Learning Representations*, 2020.

Li, Y., Yu, Y., Zhang, Q., Liang, C., He, P., Chen, W., and Zhao, T. LoSparse: Structured compression of large language models based on low-rank and sparse approximation. In *Proceedings of the 40th International Conference*

on Machine Learning, volume 202 of *Proceedings of Machine Learning Research*, pp. 20336–20350. PMLR, 2023.

Lialin, V., Muckatira, S., Shivagunde, N., and Rumshisky, A. ReLoRA: High-rank training through low-rank updates. In *International Conference on Learning Representations*, 2024.

Liu, S.-Y., Wang, C.-Y., Yin, H., Molchanov, P., Wang, Y.-C. F., Cheng, K.-T., and Chen, M.-H. DoRA: Weight-decomposed low-rank adaptation. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. Learning word vectors for sentiment analysis. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pp. 142–150, 2011.

Mozaffari, M., Yazdanbakhsh, A., Zhang, Z., and Mehri Dehnavi, M. SLoPe: Double-pruned sparse plus lazy low-rank adapter pretraining of LLMs, 2024.

Nilsback, M.-E. and Zisserman, A. Automated flower classification over a large number of classes. *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*, 2008.

Parkhi, O. M., Vedaldi, A., Zisserman, A., and Jawahar, C. V. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3498–3505, 2012.

Peng, H., Pappas, N., Yogatama, D., Schwartz, R., Smith, N. A., and Kong, L. Random feature attention. In *International Conference on Learning Representations*, 2021.

Roy, A., Saffar, M., Vaswani, A., and Grangier, D. Efficient content-based sparse attention with routing transformers. *Transactions of the Association for Computational Linguistics*, 9:53–68, 2021.

Tang, J., Zhao, Y., Zhu, K., Xiao, G., Kasikci, B., and Han, S. QUEST: Query-aware sparsity for efficient long-context LLM inference. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 47901–47911. PMLR, 2024.

Tay, Y., Dehghani, M., Abnar, S., Shen, Y., Bahri, D., Pham, P., Rao, J., Yang, L., Ruder, S., and Metzler, D. Long range arena: A benchmark for efficient transformers. *arXiv preprint arXiv:2011.04006*, 2020.

Tay, Y., Dehghani, M., Bahri, D., and Metzler, D. Efficient transformers: A survey. *ACM Computing Surveys*, 55(6): 1–28, 2022.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

Wang, S., Li, B. Z., Khabsa, M., Fang, H., and Ma, H. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

Xiao, G., Tian, Y., Chen, B., Han, S., and Lewis, M. Efficient streaming language models with attention sinks, 2024.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Xiong, Y., Zeng, Z., Chakraborty, R., Tan, M., Fung, G., Li, Y., and Singh, V. Nyströmformer: A nyström-based algorithm for approximating self-attention. In *AAAI Conference on Artificial Intelligence*, 2021.

Zaheer, M., Guruganesh, G., Dubey, K. A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L., et al. Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems*, 2020.

Zandieh, A., Han, I., Daliri, M., and Karbasi, A. KDEformer: Accelerating transformers via kernel density estimation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 40605–40623. PMLR, 2023.

Zhang, Z., Sheng, Y., Zhou, T., Chen, T., Zheng, L., Cai, R., Song, Z., Tian, Y., Ré, C., Barrett, C., Wang, Z., and Chen, B. H_2O: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems*, 2023.

Zhao, J., Zhang, Z., Chen, B., Wang, Z., Anandkumar, A., and Tian, Y. GaLore: Memory-efficient LLM training by gradient low-rank projection. In *International Conference on Machine Learning*, 2024.

Algorithm 2 RACE Low-Rank Attention Branch

1: **Input:** $Q, K, V \in \mathbb{R}^{N\times d_h}$, number of hash tables L_{s}, number of hyperplanes γ, temperature $\beta > 0$, numerical floor $\varepsilon > 0$
2: **Output:** low-rank output $O_{\mathrm{lr}} \in \mathbb{R}^{N\times d_h}$, denominator proxy $d_{\mathrm{lr}} \in \mathbb{R}^N$
3: Set $R \leftarrow 2^\gamma$ and $\mathcal{V} \leftarrow \{\pm 1\}^\gamma$
4: **for** $\ell = 1, \dots, L_{\mathrm{s}}$ **do**
5: Draw $W^{(\ell)} \in \mathbb{R}^{\gamma\times d_h}$ with i.i.d. Gaussian rows
6: Build $\Phi_Q^{(\ell)}, \Phi_K^{(\ell)} \in \mathbb{R}^{N\times R}$ with rows

$$[\phi^{(\ell)}(x)]_r = \frac{\exp\{\beta\,\tanh(W^{(\ell)}x)^\top v_r\}}{\sum_{r'=1}^{R}\exp\{\beta\,\tanh(W^{(\ell)}x)^\top v_{r'}\}}, \quad x \in \{Q_i, K_j\}.$$

7: Compute bucket mass and value summaries

$$A^{(\ell)} \leftarrow (\Phi_K^{(\ell)})^\top \mathbf{1}_N \in \mathbb{R}^R, \qquad B^{(\ell)} \leftarrow (\Phi_K^{(\ell)})^\top V \in \mathbb{R}^{R\times d_h}.$$

8: **end for**
9: Compute table-averaged numerator and denominator

$$\mathrm{Num} \leftarrow \frac{1}{L_{\mathrm{s}}}\sum_{\ell=1}^{L_{\mathrm{s}}} \Phi_Q^{(\ell)} B^{(\ell)}, \qquad \mathrm{Den} \leftarrow \frac{1}{L_{\mathrm{s}}}\sum_{\ell=1}^{L_{\mathrm{s}}} \Phi_Q^{(\ell)} A^{(\ell)}.$$

10: Return $O_{\mathrm{lr}} \leftarrow \mathrm{diag}(\mathrm{Den} + \varepsilon)^{-1}\mathrm{Num}$, $d_{\mathrm{lr}} \leftarrow \mathrm{Den}$.

Algorithm 3 Causal RACE Low-Rank Attention Branch

1: **Input:** $Q, K, V \in \mathbb{R}^{N\times d_h}$, L_{s}, γ, $\beta > 0$, chunk size C, $\varepsilon > 0$
2: **Output:** causal low-rank output $O_{\mathrm{lr}} \in \mathbb{R}^{N\times d_h}$, denominator proxy $d_{\mathrm{lr}} \in \mathbb{R}^N$
3: Build soft hash features $\Phi_Q^{(\ell)}, \Phi_K^{(\ell)} \in \mathbb{R}^{N\times R}$ for $\ell = 1, \dots, L_{\mathrm{s}}$ as in Algorithm 2 steps 3–7.
4: Pad sequence axis to $N' = \lceil N/C \rceil \cdot C$; reshape into $n_c = N'/C$ chunks of size C. Denote chunk-n slices $\Phi_{Q,n}^{(\ell)}, \Phi_{K,n}^{(\ell)} \in \mathbb{R}^{C\times R}$ and $V_n \in \mathbb{R}^{C\times d_h}$.
5: **for** each table ℓ and chunk n **do**
6: $A_n^{(\ell)} \leftarrow (\Phi_{K,n}^{(\ell)})^\top \mathbf{1}_C \in \mathbb{R}^R, \quad B_n^{(\ell)} \leftarrow (\Phi_{K,n}^{(\ell)})^\top V_n \in \mathbb{R}^{R\times d_h}$.
7: **end for**
8: Compute strict prefix states via cumulative sum and one-chunk right shift: $\mathrm{sA}_n^{(\ell)} \leftarrow \sum_{m<n} A_m^{(\ell)}$, $\mathrm{sB}_n^{(\ell)} \leftarrow \sum_{m<n} B_m^{(\ell)}$.
9: **Inter-chunk** (queries read from all prior chunks):

$$\mathrm{inum}_n \leftarrow \tfrac{1}{L_{\mathrm{s}}}\textstyle\sum_\ell \Phi_{Q,n}^{(\ell)}\mathrm{sB}_n^{(\ell)}, \qquad \mathrm{iden}_n \leftarrow \tfrac{1}{L_{\mathrm{s}}}\textstyle\sum_\ell \Phi_{Q,n}^{(\ell)}\mathrm{sA}_n^{(\ell)}.$$

10: **Intra-chunk** (causally-masked soft attention within chunk):

$$M_n \leftarrow \tfrac{1}{L_{\mathrm{s}}}\textstyle\sum_\ell \Phi_{Q,n}^{(\ell)}(\Phi_{K,n}^{(\ell)})^\top \odot \mathrm{tril}(\mathbf{1}_{C\times C}),$$

$\mathrm{anum}_n \leftarrow M_n V_n, \quad \mathrm{aden}_n \leftarrow M_n \mathbf{1}_C$.
11: Combine: $d_{\mathrm{lr},n} \leftarrow \mathrm{iden}_n + \mathrm{aden}_n, \quad O_{\mathrm{lr},n} \leftarrow \mathrm{diag}(d_{\mathrm{lr},n} + \varepsilon)^{-1}(\mathrm{inum}_n + \mathrm{anum}_n)$.
12: Concatenate chunks, remove padding, and return $O_{\mathrm{lr}}, d_{\mathrm{lr}}$.

A. Branch Algorithms for ELSAA

This appendix gives full pseudocode for all five branch-level procedures used by ELSAA: the non-causal RACE low-rank branch, its causal extension, the non-causal sortLSH sparse branch, its causal recursive extension, and the complete causal ELSAA fusion procedure.

B. Proofs and Additional Rank Corollaries

We prove the rank statements from Section 4. The proof has four ingredients. First, we relate the angular collision probabilities q_{ij} to simpler lower and upper bounds. Second, a generic rank-r matrix can increase the rank of a fixed matrix by r, unless full rank is already reached. Third, the generic rank of a sparse matrix equals the maximum matching size of its

Algorithm 4 sortLSH Sparse Exact Attention Branch

Input: $Q, K, V \in \mathbb{R}^{N \times d_h}$, LSH map $\mathcal{H}(\cdot)$, block size b, $\varepsilon > 0$
Output: sparse output $O_{\text{sparse}} \in \mathbb{R}^{N \times d_h}$, denominator $d_{\text{sparse}} \in \mathbb{R}^N$
Hash: $h_i^Q \leftarrow \mathcal{H}(Q_i)$, $h_j^K \leftarrow \mathcal{H}(K_j)$.
Sort: let $P_Q, P_K \in \text{Sym}(N)$ sort h^Q, h^K in non-decreasing order; form Q_s, K_s, V_s accordingly.
Partition sorted sequence into consecutive blocks $\{B_t\}$ of size b.
for each block B_t **do**
  $Z_t \leftarrow Q_s[B_t]K_s[B_t]^\top/\sqrt{d_h}, \quad A_t \leftarrow \exp(Z_t), \quad d_s[B_t] \leftarrow A_t\mathbf{1}$.
  $O_s[B_t] \leftarrow \text{diag}(d_s[B_t] + \varepsilon)^{-1} A_t V_s[B_t]$.
end for
Undo query permutation: $O_{\text{sparse}}[P_Q(t)] \leftarrow O_s[t]$, $d_{\text{sparse}}[P_Q(t)] \leftarrow d_s[t]$.
Return $O_{\text{sparse}}, d_{\text{sparse}}$.

Algorithm 5 Causal sortLSH Sparse Exact Attention Branch

Input: $Q, K, V \in \mathbb{R}^{N \times d_h}$, LSH map $\mathcal{H}(\cdot)$, block size b, base length N_0, $\varepsilon > 0$
Output: causal sparse output $O_{\text{sparse}} \in \mathbb{R}^{N \times d_h}$, $\log d_{\text{sparse}} \in \mathbb{R}^N$
Return $\textsc{CausalSparse}(Q, K, V)$.
procedure $\textsc{CausalSparse}(Q, K, V)$:
  if $n \le N_0$ **then** return exact causal attention $(O, \log d)$.
  If n odd, pad by one zero token ($n \leftarrow n + 1$); remove on return.
  $m \leftarrow n/2$; split into past $(Q_{\text{p}}, K_{\text{p}}, V_{\text{p}}) \leftarrow [\,\cdot\,]_{1:m}$ and future $(Q_{\text{f}}, K_{\text{f}}, V_{\text{f}}) \leftarrow [\,\cdot\,]_{m+1:n}$.
  $(O_{\text{top}}, \log d_{\text{top}}) \leftarrow \textsc{CausalSparse}(Q_{\text{p}}, K_{\text{p}}, V_{\text{p}})$ {past–past causal}
  $(O_{\text{diag}}, \log d_{\text{diag}}) \leftarrow \textsc{CausalSparse}(Q_{\text{f}}, K_{\text{f}}, V_{\text{f}})$ {future–future causal}
  $(O_{\text{off}}, \log d_{\text{off}}) \leftarrow \textsc{SortLSH}(Q_{\text{f}}, K_{\text{p}}, V_{\text{p}}; b)$ {future–past, non-causal}
  $(O_{\text{bot}}, \log d_{\text{bot}}) \leftarrow \textsc{MergeLSE}(O_{\text{diag}}, \log d_{\text{diag}}, O_{\text{off}}, \log d_{\text{off}})$
  Return $\text{concat}(O_{\text{top}}, O_{\text{bot}})$, $\text{concat}(\log d_{\text{top}}, \log d_{\text{bot}})$.
procedure $\textsc{SortLSH}(Q, K, V; b)$: apply Algorithm 4 and return $(O, \log d_{\text{sparse}})$.
procedure $\textsc{MergeLSE}(O_1, \log d_1, O_2, \log d_2)$:
  $\mu \leftarrow \max(\log d_1, \log d_2)$; $w_k \leftarrow \exp(\log d_k - \mu)$; $s \leftarrow w_1 + w_2$.
  Return $O \leftarrow (w_1 O_1 + w_2 O_2)/(s + \varepsilon)$, $\log d \leftarrow \mu + \log(s + \varepsilon)$.

Algorithm 6 Causal ELSAA: Efficient Low-Rank and Sparse Approximation of Attention

Input: hidden states $X \in \mathbb{R}^{N \times d}$, shared projections W_Q, W_K, W_V, W_O, RACE parameters $L_{\text{s}}, \gamma, \beta, C$, sortLSH parameters b, N_0, gate network G_θ, coefficient $\lambda_i > 0$, $\varepsilon > 0$
Output: causal hybrid attention output $O \in \mathbb{R}^{N \times d}$
$Q \leftarrow XW_Q, \quad K \leftarrow XW_K, \quad V \leftarrow XW_V$.
$(O_{\text{lr}}, d_{\text{lr}}) \leftarrow \textsc{CausalRACE}(Q, K, V; L_{\text{s}}, \gamma, \beta, C, \varepsilon)$ {Algorithm 3}
$(O_{\text{sparse}}, \log d_{\text{sparse}}) \leftarrow \textsc{CausalSparse}(Q, K, V; b, N_0, \varepsilon)$ {Algorithm 5}
Compute $m_{\text{sparse},i}$ via numerically stable log-sum-exp:

$$\log D_i \leftarrow \text{logsumexp}\big(\log d_{\text{sparse},i},\ \log \lambda_i + \log d_{\text{lr},i},\ \log \varepsilon\big),$$

$$m_{\text{sparse},i} \leftarrow \exp(\log d_{\text{sparse},i} - \log D_i), \qquad i = 1, \ldots, N.$$

$(g_{\text{sparse}}, g_{\text{lr}}) \leftarrow \sigma(G_\theta(X)) \in (0, 1)^{N \times 2}$.
$O_{\text{head}} \leftarrow g_{\text{sparse}} \odot m_{\text{sparse}} \odot O_{\text{sparse}} + g_{\text{lr}} \odot O_{\text{lr}}$.
Return $O \leftarrow O_{\text{head}} W_O$.

support graph. Fourth, Hall's theorem and concentration inequalities control the probability that the sparse collision graph has matching deficiency larger than r.

B.1. Generic low-rank completion

Lemma B.1 (Generic rank-r completion)**.** *Let $S \in \mathbb{R}^{n \times n}$ be fixed with*

$$\text{rank}(S) = k.$$

Let $B \in \mathbb{R}^{n\times r}$ and $A \in \mathbb{R}^{r\times n}$ be drawn from an absolutely continuous distribution. Then

$$\operatorname{rank}(S + BA) = \min\{n, k + r\} \qquad \text{almost surely.} \tag{7}$$

In particular, if $k \geq n - r$, then

$$\operatorname{rank}(S + BA) = n \qquad \text{almost surely.} \tag{8}$$

Proof. Let

$$m := \min\{n, k + r\}.$$

Since $\operatorname{rank}(S) = k$, there exist invertible matrices $P, Q \in \mathbb{R}^{n\times n}$ such that

$$PSQ = \begin{pmatrix} I_k & 0 \\ 0 & 0 \end{pmatrix}. \tag{9}$$

Rank is invariant under multiplication by invertible matrices, hence

$$\operatorname{rank}(S + BA) = \operatorname{rank}(PSQ + PBAQ). \tag{10}$$

Define

$$\widetilde{B} := PB, \qquad \widetilde{A} := AQ.$$

Because P and Q are invertible and A, B are drawn from absolutely continuous distributions, $\widetilde{A}, \widetilde{B}$ are also absolutely continuous.

We show that some $m \times m$ minor of

$$PSQ + \widetilde{B}\widetilde{A}$$

is not the zero polynomial in the entries of $\widetilde{A}, \widetilde{B}$. Let

$$t := m - k.$$

Then $0 \leq t \leq r$. Choose a deterministic value of $\widetilde{B}, \widetilde{A}$ as follows:

$$\widetilde{B}_{\{k+1,\ldots,k+t\},\{1,\ldots,t\}} = I_t, \qquad \widetilde{A}_{\{1,\ldots,t\},\{k+1,\ldots,k+t\}} = I_t,$$

and set all other entries of $\widetilde{B}, \widetilde{A}$ equal to zero. Then $\widetilde{B}\widetilde{A}$ places an identity block on coordinates $k + 1, \ldots, k + t$. Therefore

$$PSQ + \widetilde{B}\widetilde{A}$$

contains an $m \times m$ identity block, and hence has rank at least m.

Thus, at least one $m \times m$ determinant polynomial is not identically zero. Since the zero set of a nonzero polynomial has Lebesgue measure zero, the same minor is nonzero almost surely for absolutely continuous $\widetilde{A}, \widetilde{B}$. Hence

$$\operatorname{rank}(S + BA) \geq m \qquad \text{almost surely.} \tag{11}$$

On the other hand, by subadditivity of rank,

$$\operatorname{rank}(S + BA) \leq \operatorname{rank}(S) + \operatorname{rank}(BA) \leq k + r. \tag{12}$$

Also $\operatorname{rank}(S + BA) \leq n$. Therefore

$$\operatorname{rank}(S + BA) \leq \min\{n, k + r\} = m. \tag{13}$$

Combining Equations (11) and (13) gives

$$\operatorname{rank}(S + BA) = m = \min\{n, k + r\}$$

almost surely. □

B.2. Structural rank and maximum matchings

Definition B.2 (Structural rank)**.** For a support pattern $\Omega \subseteq [n] \times [n]$, define the bipartite graph

$$G_\Omega = ([n]_{\text{row}}, [n]_{\text{col}}, \Omega).$$

The structural rank of Ω is the largest rank achievable by any matrix whose nonzero entries are restricted to the support Ω.

Lemma B.3 (Structural rank equals maximum matching)**.** *Let $X_\Omega \in \mathbb{R}^{n\times n}$ be a sparse matrix supported on Ω. Assume that its nonzero entries are algebraically generic. Then*

$$\operatorname{rank}(X_\Omega) = \nu(\Omega) \qquad \text{almost surely.} \tag{14}$$

Proof. Let

$$m := \nu(\Omega).$$

Since the graph G_Ω has a matching of size m, there exist row and column sets $I, J \subseteq [n]$, with $|I| = |J| = m$, such that the subgraph induced by I, J contains a perfect matching.

Consider the determinant of the submatrix $X_{\Omega,I,J}$. Expanding this determinant as a polynomial in the nonzero variables

$$\{x_{ij} : (i,j) \in \Omega\},$$

the perfect matching contributes a monomial of the form

$$\prod_{(i,j)\in M} x_{ij}. \tag{15}$$

Different perfect matchings correspond to different monomials. Therefore this determinant polynomial is not identically zero. Since the entries are algebraically generic, the determinant is nonzero almost surely. Hence

$$\operatorname{rank}(X_\Omega) \geq m. \tag{16}$$

Conversely, suppose that $\operatorname{rank}(X_\Omega) \geq m+1$. Then there exists an $(m+1)\times(m+1)$ minor that is not identically zero as a polynomial in the nonzero entries. In the determinant expansion of this minor, at least one permutation monomial must be present. Such a monomial corresponds to a matching of size $m+1$ in G_Ω, contradicting the maximality of $\nu(\Omega) = m$. Therefore

$$\operatorname{rank}(X_\Omega) \leq m. \tag{17}$$

Combining Equations (16) and (17) gives

$$\operatorname{rank}(X_\Omega) = \nu(\Omega)$$

almost surely. □

B.3. Hall deficiency and weighted Hall failure

Definition B.4 (Neighborhood of a row set)**.** For a set $I \subseteq [n]$ of row vertices, define its neighborhood in the support graph G_Ω by

$$N_\Omega(I) := \{j \in [n] : \exists i \in I \text{ such that } (i,j) \in \Omega\}. \tag{18}$$

Lemma B.5 (Deficiency form of Hall's theorem)**.** *For the bipartite graph G_Ω,*

$$n - \nu(\Omega) = \max_{I\subseteq[n]} \left(|I| - |N_\Omega(I)|\right). \tag{19}$$

Consequently,

$$\nu(\Omega) < n - r$$

if and only if there exists $I \subseteq [n]$ such that

$$|N_\Omega(I)| \leq |I| - r - 1. \tag{20}$$

Proof. This is the standard deficiency form of Hall's theorem. A matching of size at least $n - r$ exists if and only if every row set $I \subseteq [n]$ has deficiency at most r, namely

$$|I| - |N_\Omega(I)| \le r.$$

Equivalently,

$$|N_\Omega(I)| \ge |I| - r \qquad \text{for all } I \subseteq [n].$$

Thus, the condition fails if and only if there exists $I \subseteq [n]$ such that

$$|N_\Omega(I)| \le |I| - r - 1.$$

□

Lemma B.6 (Weighted Hall failure bound)**.** *Assume that the edge indicators Ω_{ij} are independent Bernoulli random variables with probabilities q_{ij}. Let $R := r + 1$. For $I, U \subseteq [n]$, define*

$$\Lambda(I, U) := \sum_{i \in I,\ j \in U} q_{ij}.$$

Then

$$\mathbb{P}[\nu(\Omega) < n - r] \le \sum_{\substack{I, U \subseteq [n] \\ |I| \ge R,\ |U| \ge R,\ |I| + |U| \ge n + R}} \exp(-\Lambda(I, U)). \tag{21}$$

Proof. By Theorem B.5, the event $\nu(\Omega) < n - r$ occurs if and only if there exists $I \subseteq [n]$ such that

$$|N_\Omega(I)| \le |I| - r - 1.$$

Let $J := N_\Omega(I)$ and $U := J^c$. Then no edges exist from I to U. Moreover,

$$|U| = n - |J| \ge n - |I| + r + 1.$$

Equivalently,

$$|I| + |U| \ge n + r + 1 = n + R.$$

Also, such a violation can occur only when $|I| \ge R$, and the previous display implies $|U| \ge R$. Thus, Hall failure implies the existence of sets $I, U \subseteq [n]$ satisfying

$$|I| \ge R, \qquad |U| \ge R, \qquad |I| + |U| \ge n + R,$$

with no edges in the rectangle $I \times U$.

For fixed I, U, the probability that there are no edges in $I \times U$ is

$$\prod_{i \in I,\ j \in U} (1 - q_{ij}) \le \exp\left(- \sum_{i \in I,\ j \in U} q_{ij} \right) = \exp(-\Lambda(I, U)).$$

Taking a union bound over all admissible I, U gives the result. □

B.4. Proof of the deterministic rank proposition

Proof of Theorem 4.5. By Theorem B.3, the sparse matrix S_Ω satisfies

$$\operatorname{rank}(S_\Omega) = \nu(\Omega)$$

almost surely under the generic-values assumption. Applying Theorem B.1 with $S = S_\Omega$, we get

$$\begin{aligned} \operatorname{rank}(S_\Omega + BA) &= \min\{n, \operatorname{rank}(S_\Omega) + r\} \\ &= \min\{n, \nu(\Omega) + r\} \end{aligned} \tag{22}$$

almost surely. In particular, if

$$\nu(\Omega) \ge n - r,$$

then

$$\operatorname{rank}(S_\Omega + BA) = n$$

almost surely. □

B.5. Proof of the main full-rank theorem

Proof of Theorem 4.6. Condition on Q, K. Then the angular collision probabilities q_{ij} are fixed numbers in $[0, 1]$. By Theorem B.6,

$$\mathbb{P}[\nu(\Omega) < n - r \mid Q, K] \leq \Delta_r(Q, K).$$

Therefore, with probability at least $1 - \Delta_r(Q, K)$, we have

$$\nu(\Omega) \geq n - r.$$

On this event, Theorem 4.5 implies

$$\operatorname{rank}(S_\Omega + BA) = n$$

almost surely over the generic low-rank factors A, B. Hence

$$\mathbb{P}[\operatorname{rank}(S_\Omega + BA) = n \,|\, Q, K] \geq 1 - \Delta_r(Q, K).$$

□

C. Interpretable High-Probability Corollaries

The main theorem gives the full-rank probability through the quantity $\Delta_r(Q, K)$. In this section, we give several sufficient conditions under which $\Delta_r(Q, K) \leq n^{-c}$, so that

$$\mathbb{P}[\operatorname{rank}(S_\Omega + BA) = n \,|\, Q, K] \geq 1 - n^{-c}.$$

These corollaries are not meant to exhaust all possible conditions. Rather, they show different ways to interpret the rank guarantee: through uniform cut-density, through a conservative worst-case angular lower bound, and through an idealized isotropic mean model.

C.1. A combinatorial rectangle bound

Lemma C.1 (A binomial rectangle bound)**.** *Let $R \in [n]$. Suppose $a, h \in \{R, \ldots, n\}$ satisfy*

$$a + h \geq n + R.$$

Then

$$\log \binom{n}{a} + \log \binom{n}{h} \leq 8 \, \frac{ah}{n} \log \frac{en}{R}. \tag{23}$$

Proof. We use the standard bound

$$\binom{n}{t} = \binom{n}{n-t} \leq \left(\frac{en}{\min\{t, n-t\}} \right)^{\min\{t, n-t\}},$$

with the convention that the corresponding term is zero when $\min\{t, n-t\} = 0$.

First suppose $a \leq n/2$. Since $a + h \geq n + R$, we have

$$n - h \leq a - R \leq a.$$

Therefore,

$$\begin{aligned} \log \binom{n}{a} + \log \binom{n}{h} &= \log \binom{n}{a} + \log \binom{n}{n-h} \\ &\leq a \log \frac{en}{a} + (n-h) \log \frac{en}{n-h} \\ &\leq 2a \log \frac{en}{a} \\ &\leq 2a \log \frac{en}{R}. \end{aligned} \tag{24}$$

Also $h \geq n/2$, so $ah/n \geq a/2$. Hence

$$\log \binom{n}{a} + \log \binom{n}{h} \leq 4\,\frac{ah}{n} \log \frac{en}{R}.$$

The case $h \leq n/2$ is symmetric.

It remains to consider the case $a > n/2$ and $h > n/2$. Then

$$\binom{n}{a}\binom{n}{h} \leq 2^{2n},$$

so

$$\log \binom{n}{a} + \log \binom{n}{h} \leq 2n \log 2.$$

Since $a > n/2$ and $h > n/2$, we have $ah/n > n/4$. Also $\log(en/R) \geq 1$. Therefore

$$8\,\frac{ah}{n} \log \frac{en}{R} \geq 2n \geq 2n \log 2.$$

Combining the cases proves the claim. □

C.2. Deficiency-aware collision density

Corollary C.2 (Deficiency-aware collision density)**.** *Fix $c > 0$, and suppose $0 \leq r < n$. Let*

$$R := r + 1,$$

and define

$$p_{r,c} := \frac{1}{n}\left(8 \log \frac{en}{R} + \frac{(c+2)\log n}{R}\right). \tag{25}$$

Suppose that for every pair of sets $I, U \subseteq [n]$ satisfying

$$|I| \geq R, \qquad |U| \geq R, \qquad |I| + |U| \geq n + R,$$

we have the cut-density lower bound

$$\Lambda(I, U) = \sum_{i \in I,\ j \in U} q_{ij} \geq p_{r,c}|I||U|. \tag{26}$$

Then, under the assumptions of Theorem 4.6,

$$\mathbb{P}[\operatorname{rank}(S_\Omega + BA) = n \mid Q, K] \geq 1 - n^{-c}. \tag{27}$$

Proof. Let

$$R := r + 1.$$

Recall from Theorem 4.6 that the failure probability is bounded by

$$\Delta_r(Q, K) = \sum_{\substack{I, U \subseteq [n] \\ |I| \geq R,\ |U| \geq R,\ |I| + |U| \geq n + R}} \exp(-\Lambda(I, U)).$$

For fixed cardinalities

$$a := |I|, \qquad h := |U|,$$

the cut-density assumption gives

$$\Lambda(I, U) \geq p_{r,c}\, ah.$$

Therefore,

$$\Delta_r(Q,K) \le \sum_{\substack{a,h\in\{R,\dots,n\}\\ a+h\ge n+R}} \binom{n}{a}\binom{n}{h}\exp(-p_{r,c}ah). \tag{28}$$

By the definition of $p_{r,c}$,

$$p_{r,c}ah = \frac{ah}{n}\left(8\log\frac{en}{R} + \frac{(c+2)\log n}{R}\right).$$

Using Theorem C.1,

$$\log\binom{n}{a} + \log\binom{n}{h} \le 8\,\frac{ah}{n}\log\frac{en}{R}.$$

Hence, for each admissible cardinality pair (a,h),

$$\binom{n}{a}\binom{n}{h}\exp(-p_{r,c}ah) \le \exp\left(-\frac{ah}{nR}(c+2)\log n\right). \tag{29}$$

Since $a,h \ge R$ and $a+h \ge n+R$, the product ah is minimized at $(a,h)=(R,n)$ or (n,R). Thus

$$ah \ge nR.$$

Therefore each admissible cardinality pair contributes at most

$$n^{-(c+2)}.$$

There are at most n^2 admissible pairs (a,h). Consequently,

$$\Delta_r(Q,K) \le n^2 \cdot n^{-(c+2)} = n^{-c}.$$

Applying Theorem 4.6 completes the proof. □

The threshold $p_{r,c}$ is deficiency-aware. When $r=0$, the sparse graph itself must contain a perfect matching, and the sufficient sparse density has the familiar $O(\log n/n)$ scaling. When $r>0$, the sparse graph is allowed to have matching deficiency up to r, and the rank-r low-rank branch fills the missing directions.

C.3. Conservative worst-case angular condition

Corollary C.3 (Conservative worst-case angular condition)**.** *Fix $c>0$, suppose $0 \le r < n$, and let $p_{r,c}$ be defined as in Equation (25). Let*

$$\rho_{\min} := \min_{i,j}\rho_{ij}.$$

If

$$1-\exp(-L_{\mathrm{s}}\rho_{\min}^{\gamma}) \ge p_{r,c}, \tag{30}$$

then, under the assumptions of Theorem 4.6,

$$\mathbb{P}[\operatorname{rank}(S_\Omega + BA) = n \,|\, Q,K] \ge 1-n^{-c}. \tag{31}$$

Equivalently, when $p_{r,c}<1$ and $\rho_{\min}>0$, it is sufficient that

$$L_{\mathrm{s}} \ge \frac{-\log(1-p_{r,c})}{\rho_{\min}^{\gamma}}. \tag{32}$$

Proof. For every pair (i,j),

$$q_{ij} \ge 1-\exp(-L_{\mathrm{s}}\rho_{ij}^{\gamma}).$$

By definition of $\rho_{\min}$,

$$\rho_{ij} \ge \rho_{\min} \qquad \text{for all } i,j.$$

Therefore,

$$q_{ij} \geq 1 - \exp(-L_{\mathrm{s}}\rho_{\min}^{\gamma}).$$

If

$$1 - \exp(-L_{\mathrm{s}}\rho_{\min}^{\gamma}) \geq p_{r,c},$$

then $q_{ij} \geq p_{r,c}$ for all i, j. Hence, for every admissible rectangle $I \times U$,

$$\Lambda(I, U) = \sum_{i \in I,\ j \in U} q_{ij} \geq p_{r,c}|I||U|.$$

Thus the cut-density condition in Theorem C.2 holds, and the desired high-probability full-rank conclusion follows.

Finally, when $p_{r,c} < 1$ and $\rho_{\min} > 0$, the inequality

$$1 - \exp(-L_{\mathrm{s}}\rho_{\min}^{\gamma}) \geq p_{r,c}$$

is equivalent to

$$\exp(-L_{\mathrm{s}}\rho_{\min}^{\gamma}) \leq 1 - p_{r,c},$$

which holds whenever

$$L_{\mathrm{s}} \geq \frac{-\log(1 - p_{r,c})}{\rho_{\min}^{\gamma}}.$$

□

Remark C.4 (Worst-case angular conditions are pessimistic). Theorem C.3 uses the minimum angular similarity over all n^2 query-key pairs and is therefore conservative. In practice, full rank does not require every pair to have large collision probability; it only requires that no Hall-relevant rectangle has too little total collision mass. Thus, some individual pairs may have very small collision probability, as long as no large query set is separated from too many key vertices.

C.4. Isotropic mean Hoeffding certificate

Corollary C.5 (Isotropic mean Hoeffding certificate)**.** *Fix $c > 0$, $\delta \in (0, 1)$, and suppose $0 \leq r < n$. Let*

$$R := r + 1.$$

For $a, h \in \{R, \ldots, n\}$ satisfying $a + h \geq n + R$, define

$$\tau_{a,h,c} := \frac{\log\binom{n}{a} + \log\binom{n}{h} + (c+2)\log n}{ah}, \tag{33}$$

and

$$\beta_{a,h,\delta} := \sqrt{\frac{\log\binom{n}{a} + \log\binom{n}{h} + \log(n^2/\delta)}{2ah}}. \tag{34}$$

Let

$$\Theta_{r,c,\delta} := \max_{\substack{a,h \in \{R,\ldots,n\} \\ a+h \geq n+R}} \left(\tau_{a,h,c} + \beta_{a,h,\delta}\right). \tag{35}$$

Assume an idealized isotropic collision-probability model in which the entries $q_{ij} \in [0, 1]$ are independent random variables with common mean μ_q. Moreover, assume that the isotropic angular estimate is a conservative lower approximation to this mean:

$$\mu_q \geq \bar{q}_{\mathrm{iso}} - \varepsilon_{\mathrm{iso}}, \qquad \bar{q}_{\mathrm{iso}} := 1 - \left(1 - 2^{-\gamma}\right)^{L_{\mathrm{s}}}, \tag{36}$$

where $\varepsilon_{\mathrm{iso}} \geq 0$ measures the approximation slack. In the sparse-collision regime $L_{\mathrm{s}}2^{-\gamma} \ll 1$,

$$\bar{q}_{\mathrm{iso}} = L_{\mathrm{s}}2^{-\gamma} + O\left(L_{\mathrm{s}}^2 2^{-2\gamma}\right).$$

If

$$\bar{q}_{\mathrm{iso}} - \varepsilon_{\mathrm{iso}} \geq \Theta_{r,c,\delta}, \tag{37}$$

then, with probability at least $1-\delta$ over the draw of the probability matrix (q_{ij}), all Hall-relevant rectangles satisfy

$$\frac{1}{|I||U|}\sum_{i\in I,\ j\in U} q_{ij} \geq \tau_{|I|,|U|,c}. \tag{38}$$

Consequently, under Theorems 4.3 and 4.4, with probability at least $1-\delta$ over the draw of (q_{ij}),

$$\mathbb{P}[\mathrm{rank}(S_\Omega + BA) = n \,|\, (q_{ij})_{i,j}] \geq 1 - n^{-c}. \tag{39}$$

Equivalently, over the joint randomness of the collision probabilities, the sparse support, and the low-rank factors,

$$\mathbb{P}[\mathrm{rank}(S_\Omega + BA) = n] \geq 1 - \delta - n^{-c}. \tag{40}$$

Proof. Fix admissible cardinalities $a, h \in \{R, \dots, n\}$ satisfying

$$a + h \geq n + R.$$

For fixed sets $I, U \subseteq [n]$ with $|I| = a$ and $|U| = h$, define

$$\overline{q}(I,U) := \frac{1}{ah}\sum_{i\in I,\ j\in U} q_{ij}.$$

Under the idealized isotropic collision-probability model, the entries $q_{ij} \in [0,1]$ are independent with common mean μ_q. Hence Hoeffding's inequality gives

$$\mathbb{P}\left[\overline{q}(I,U) < \mu_q - \beta_{a,h,\delta}\right] \leq \exp(-2ah\beta^2_{a,h,\delta}).$$

By the definition of $\beta_{a,h,\delta}$,

$$\exp(-2ah\beta^2_{a,h,\delta}) = \exp\left(-\log\binom{n}{a} - \log\binom{n}{h} - \log(n^2/\delta)\right).$$

Taking a union bound over all $\binom{n}{a}\binom{n}{h}$ pairs of sets with these cardinalities, the probability that any such rectangle violates the bound is at most

$$\frac{\delta}{n^2}.$$

Taking another union bound over at most n^2 admissible cardinality pairs (a,h), we obtain that, with probability at least $1-\delta$, every Hall-relevant rectangle satisfies

$$\overline{q}(I,U) \geq \mu_q - \beta_{|I|,|U|,\delta}.$$

By assumption,

$$\mu_q \geq \bar{q}_{\mathrm{iso}} - \varepsilon_{\mathrm{iso}} \geq \Theta_{r,c,\delta}.$$

Since

$$\Theta_{r,c,\delta} = \max_{\substack{a,h\in\{R,\dots,n\}\\ a+h\geq n+R}} \left(\tau_{a,h,c} + \beta_{a,h,\delta}\right),$$

it follows that every Hall-relevant rectangle satisfies

$$\overline{q}(I,U) \geq \tau_{|I|,|U|,c}.$$

Equivalently,

$$\Lambda(I,U) = \sum_{i\in I,\ j\in U} q_{ij} \geq |I||U|\tau_{|I|,|U|,c}.$$

By the definition of $\tau_{a,h,c}$,

$$\Lambda(I,U) \geq \log\binom{n}{|I|} + \log\binom{n}{|U|} + (c+2)\log n.$$

Therefore,

$$\begin{aligned}\Delta_r &= \sum_{\substack{I,U\subseteq[n]\\ |I|\ge R,\ |U|\ge R,\ |I|+|U|\ge n+R}} \exp(-\Lambda(I,U)) \\ &\le \sum_{\substack{a,h\in\{R,\dots,n\}\\ a+h\ge n+R}} \binom{n}{a}\binom{n}{h} \exp\left(-\log\binom{n}{a} - \log\binom{n}{h} - (c+2)\log n\right) \\ &\le n^2 \cdot n^{-(c+2)} = n^{-c}. \end{aligned} \tag{41}$$

Applying Theorem 4.6 conditionally on the realized probability matrix (q_{ij}) gives

$$\mathbb{P}[\operatorname{rank}(S_\Omega + BA) = n \mid (q_{ij})_{i,j}] \ge 1 - n^{-c}$$

with probability at least $1-\delta$ over the draw of (q_{ij}).

Finally, by the union bound over the two sources of failure, namely failure of the probability matrix to satisfy the rectangle-average condition and failure of the sampled sparse graph to have matching size at least $n-r$, the joint success probability is at least

$$1 - \delta - n^{-c}.$$

□

The role of δ in Theorem C.5 is to control the probability that the random collision-probability matrix itself has a low-density Hall-relevant cut. The term n^{-c} controls the subsequent failure probability of the sampled sparse graph, conditioned on that probability matrix. In the common sparse-collision regime, the practical design rule suggested by Equation (37) is

$$L_{\mathrm{s}} 2^{-\gamma} \gtrsim \Theta_{r,c,\delta}.$$

Thus, increasing L_{s} increases sparse coverage, while increasing γ makes collisions more selective.

C.5. Interpretation and technical remarks

Remark C.6 (Interpretation). The rank guarantee is an expressivity statement. It shows that the hybrid sparse + low-rank construction avoids the rank collapse of purely low-rank attention. The sparse collision graph contributes rank through its maximum matching, while the low-rank branch supplies r additional dense directions. Thus,

$$\text{sparse matching size} + \text{low-rank dimension} \ge n$$

is sufficient for the hybrid attention matrix to be full rank.

Remark C.7 (Scope of the probabilistic assumptions). The corollaries above should be read as interpretable regimes in which the error term in Theorem 4.6 becomes small. Some of these regimes make simplifying assumptions, such as independent edge sampling or an idealized isotropic model for the collision probabilities. These assumptions are not intended to fully model every practical hashing implementation. Rather, they expose the mechanism behind the algorithm: if angular sparse sampling produces enough coverage across Hall-relevant cuts, then the sparse component has matching deficiency at most r, and the rank-r low-rank component fills the remaining directions.

Remark C.8 (Independence). Theorem 4.6 is stated under an independent angular edge model. Practical LSH collisions may be dependent because the same hash functions are reused across multiple query-key pairs. The deterministic implication in Theorem 4.5, however, does not require independence. It applies to any realized support Ω satisfying $\nu(\Omega) \ge n-r$. Independence is used only to control the probability that this matching condition fails.

Remark C.9 (Deterministic version). The deterministic implication

$$\nu(\Omega) \ge n - r \quad \Longrightarrow \quad \operatorname{rank}(S_\Omega + BA) = n$$

holds under the generic sparse-values and generic low-rank assumptions, and does not require independence of the sparse support. Independence is used only to upper bound the probability that $\nu(\Omega) < n-r$.

Table 6. Main dataset-level hyperparameters. Here N denotes the input sequence length after tokenization or patchification.

Dataset / Task	N	Layers	Heads	d	MLP dim	Batch	Grad. accum.	LR	WD	Epochs
IMDB	512	1	2	128	512	32	1	1×10^{-5}	5×10^{-5}	150
Fashion-MNIST	784	2	4	384	1536	32	1–2	6×10^{-4}	0.1	150
Oxford-IIIT Pet	16,384	2	4	384	1536	32	1–2	6×10^{-4}	0.1	150
Flowers-102	16,384	2	4	384	1536	32	1–2	6×10^{-4}	0.1	150
Food-101	16,384	8	8	512	2048	8	4	3×10^{-4}	0.001	100
ArXiv classification	32,000	4	4	256	1024	8	16	3×10^{-4}	0.01	33
Text Retrieval	64,000	4	4	256	1024	4	16	3×10^{-4}	0.01	50

Remark C.10 (When exact sparse values are not generic). If the sparse exact values $S_\Omega = \Omega \odot M^\star$ are deterministic and not assumed to be generic, one can use the deterministic condition

$$\operatorname{rank}(S_\Omega) \geq n - r.$$

Then Theorem B.1 alone implies

$$\operatorname{rank}(S_\Omega + BA) = n$$

almost surely over the low-rank factors A, B. The matching condition $\nu(\Omega) \geq n - r$ is a support-level sufficient condition for this rank condition under generic sparse values.

Remark C.11 (Independent safety sparsifier). If one wants a literal independent-edge guarantee while preserving a practical LSH collision rule, one may augment the LSH support by an independent safety sparsifier:

$$\Omega = \Omega_{\mathrm{LSH}} \cup \Omega_{\mathrm{safe}}, \qquad \Omega_{\mathrm{safe},ij} \sim \mathrm{Bernoulli}(p_{r,c})\,.$$

By Theorem C.2, the safety support alone is sufficient to guarantee

$$\operatorname{rank}(S_\Omega + BA) = n$$

with probability at least $1 - n^{-c}$, under the generic sparse-values and generic low-rank assumptions. Its expected number of additional edges is

$$n^2 p_{r,c} = n\left(8 \log \frac{en}{r+1} + \frac{(c+2)\log n}{r+1}\right).$$

D. Experiment Hyperparameters

We evaluate all attention variants using encoder-style, non-causal Transformer architectures. Unless otherwise stated, we train with cross-entropy loss and the AdamW optimizer. We use dropout 0.1, set `qkv_bias=False`, and fix the random seed to 42. For the hybrid methods, the gate is a two-layer MLP with SiLU activation and two independent sigmoid outputs, one for the sparse branch and one for the RACE branch. We do not normalize the gates.

For the image experiments, Fashion-MNIST uses 28×28 grayscale images with patch size 1, giving $N = 784$. Oxford-IIIT Pet and Flowers-102 use 512×512 RGB images with patch size 4, giving $128 \times 128 = 16{,}384$ image tokens. For Food-101, we use the long-image setting with 512×512 inputs, patch size 4, and $N = 16{,}384$. The vision models use a learnable class token and learnable positional embeddings.

For ArXiv classification, we tokenize documents using a basic English tokenizer, keep documents with at least 1,000 raw tokens, and perform class-balanced streaming packing. Documents from the same class are concatenated until the target sequence length is reached. For the reported ArXiv classification setting, we use $N = 32{,}000$, a vocabulary limit of 50,000, and packed train/test examples produced from the class-balanced long-document subset. For Text Retrieval, we construct binary retrieval-pair examples from the packed ArXiv documents. Each input is formatted as

$$[\mathrm{CLS}]\ \mathrm{doc}_a\ [\mathrm{SEP}]\ \mathrm{doc}_b,$$

with label 1 if the two documents come from the same ArXiv class and label 0 otherwise. We use 4,000 training pairs and 1,000 test pairs at $N = 64{,}000$.

Table 7. Attention-specific hyperparameters.

Component	Text settings	Vision settings
RACE hash bits γ	4 for ArXiv/Retrieval, 3 for IMDB	2
RACE tables L_s	4 for ArXiv/Retrieval, 2 for IMDB	5
RACE ensembles M	1 for ArXiv/Retrieval, 2 for IMDB	1
Sort_LSH bits	5	4–5
Sort_LSH block size s	256 for ArXiv/Retrieval, 32 for IMDB	32
Sort_LSH min length	4096 for ArXiv/Retrieval, 256 for IMDB	256
Neighbor blocks	0	0
Gate hidden dim	64	128
Gate normalization	False	False
ϵ for denominator correction	10^{-6}	10^{-6}

For ELSAA, the sparse branch is Sort_LSH and the low-rank branch is RACE. The output is computed as

$$O = g_{\text{sparse}}\, m_{\text{sparse}}\, O_{\text{sparse}} + g_{\text{race}}\, O_{\text{race}},$$

where g_{sparse} and g_{race} are token-wise sigmoid gates. In the denominator-aware version, we use

$$m_{\text{sparse}} = \frac{d_{\text{sparse}}}{d_{\text{sparse}} + \lambda d_{\text{race}} + \epsilon}.$$

For the scalar-λ version, λ is parameterized as

$$\lambda = \exp(\ell_\lambda),$$

initialized with $\lambda = 1.0$, and learned during training. For the input-dependent version, we use

$$\lambda_i = c + \sigma(w^\top q_i + b),$$

where c is initialized to 0.3, is learnable, and is constrained to be nonnegative. The bias is initialized so that the initial average target is approximately 0.8. We detach q_i from the lambda path, use $\lambda_{\min} = 10^{-6}$, and initialize w with standard deviation 10^{-3}.

For IMDB, we build a long-review subset at $N = 512$. Reviews with length between N and $2N$ are kept directly, longer reviews are split into overlapping windows with stride $N/2$, and shorter reviews from the same class are concatenated until they reach the target length. During training, we apply light EDA augmentation using random deletion with probability 0.05 or random token swaps.

For the vision datasets, we use standard data augmentation. Fashion-MNIST uses random horizontal flipping and random cropping with padding 4 during training. Oxford-IIIT Pet, Flowers-102, and Food-101 use random resized cropping and horizontal flipping for training, and resize followed by center crop for validation. RGB image datasets are normalized with ImageNet mean and standard deviation.

E. Numerical Complexity Example

Attention interaction count. We give a concrete example to illustrate the scale of the savings from the sparse–low-rank construction. For

$$N = 32{,}000, \qquad s = 256, \qquad L_{\text{s}} = 4, \qquad \gamma = 4,$$

full attention computes

$$N^2 = 1.024 \times 10^9$$

query–key interactions per head. The sparse sortLSH branch computes

$$Ns = 32{,}000 \cdot 256 = 8.192 \times 10^6$$

selected interactions, which is 0.8% of full attention. The RACE branch uses

$$NL_{\text{s}}2^\gamma = 32{,}000 \cdot 4 \cdot 16 = 2.048 \times 10^6$$

bucket interactions, which is 0.2% of full attention. Therefore, ELSAA uses

$$N(s + L_{\mathrm{s}} 2^{\gamma}) = 32{,}000(256 + 64) = 1.024 \times 10^{7}$$

interactions, which is 1.0% of full attention. This corresponds to roughly a 99.0% reduction in attention interactions compared with exact dense attention.

Hashing and sorting overhead. The hashing overhead is linear or near-linear in sequence length. The sortLSH branch requires angular hash projections and sorting, approximately

$$O(N d\, \gamma_{\mathrm{sort}}) + O(N \log N),$$

where γ_{sort} denotes the number of hash bits used by the sparse selector. The RACE branch requires soft-hash projections and bucket-feature construction, approximately

$$O(N d\, L_{\mathrm{s}} \gamma) + O(N L_{\mathrm{s}} \gamma 2^{\gamma}).$$

In our experiments, s, L_{s}, γ, and γ_{sort} are fixed hyperparameters. Therefore, these terms scale linearly or near-linearly in N, whereas dense attention scales quadratically. For sufficiently long sequences, the hashing and sorting overhead is dominated by the saved N^2 query–key computation.

Parameter overhead. The additional learnable parameters introduced by ELSAA are small compared with the base Transformer layer. The standard attention projections

$$W_Q, W_K, W_V, W_O \in \mathbb{R}^{d \times d}$$

already contribute $O(d^2)$ parameters. In contrast, the scalar-λ version of ELSAA adds only $O(1)$ parameters, a query-dependent λ version adds $O(d)$ parameters, and the gate MLP adds $O(dg)$ parameters for gate hidden width $g \ll d$. Thus, the parameter overhead is negligible relative to the base attention projections, while the main savings come from reducing the input-dependent attention computation.