RACE ATTENTION: A LINEAR-TIME ATTENTION MECHANISM FOR LONG-SEQUENCE TRAINING WITH EXTREME-LENGTH ATTENTION-LAYER SCALING

Anonymous authors
Paper under double-blind review

ABSTRACT

Softmax Attention has a quadratic time complexity in sequence length, which becomes prohibitive to run at long contexts, even with highly optimized GPU kernels. For example, FlashAttention2 and FlashAttention3 (exact, GPU-optimized implementations of Softmax Attention) cannot complete a single forward–backward pass of a multi-head attention layer once the context exceeds ~ 4 million tokens on an NVIDIA GH200 (96 GB). We introduce RACE Attention, a kernel-inspired alternative to Softmax Attention that is strictly linear in sequence length and embedding dimension. RACE Attention replaces the exponential kernel with a sharpened angular similarity, and approximates attention outputs via randomized projections and soft Locality-Sensitive Hashing (LSH). Across language modeling, masked language modeling, and text/image classification, RACE Attention matches or outperforms strong baselines up to 64K seqeuence length while reducing wall-clock time and memory. In a controlled scale test, it processes up to 12 million tokens during a single forward-backward pass on an NVIDIA GH200 GPU and 75 million tokens on an Intel Xeon® Gold 5220R CPU—well beyond the practical limits of the current state-of-the-art attention implementations. RACE Attention thus offers a practical and theoretically grounded mechanism for long-context training on today's hardware.

1 INTRODUCTION

The Transformer (Vaswani et al., 2017; Dehghani et al., 2019) is the backbone of modern sequence modeling across language, vision Parmar et al. (2018), and speech Luo et al. (2020). We have seen remarkable improvements over the past few years in reasoning and understanding capabilities. Most of these are attributed to the increased parameters of the transformers along with the capability to process longer context windows than before. All this progress, however, rests on a computationally expensive primitive: Softmax Attention, whose time scales quadratically with context length. As models and contexts grow, spanning multi-document reasoning, long-form code, audio, and video, the quadratic barrier increasingly dictates who can train and deploy capable systems. Industrial labs mitigate the cost with large-scale distributed hardware; most practitioners cannot. There is a growing need for attention mechanisms that are *accurate*, *fast*, and *memory-efficient*. To highlight the limits of Softmax Attention: even with FlashAttention2 Dao (2023) and FlashAttention3 Shah et al. (2024), the state-of-the-art GPU implementations, a single forward–backward pass of a multi-head attention layer (1 batch, 4 heads, embedding dimension of 128) remains computationally and memory intensive and cannot process sequences beyond ~ 4 million tokens on an NVIDIA GH200 (96 GB) GPU. Clearly, to achieve an outrageously long context where the target context size is hundreds of millions of tokens or beyond, fundamental rethinking of attention will be required (Bahdanau et al., 2014).

Linearized and Low-Rank Approximations to Quadratic Attention: Due to the significance of the problem, a very large body of work attempts to accelerate attention by approximating softmax with linear approximations Zeng et al. (2021) or clever kernel feature maps (Katharopoulos et al., 2020; Choromanski et al., 2021; Peng et al., 2021; Qin et al., 2022). A work more closely related to ours is YOSO Attention Zeng et al. (2021), which also approximates a powered angular kernel using hard LSH. Despite this same kernel formulation, the underlying estimators differ substantially. RACE Attention leverages a smooth, differentiable relaxation of classical RACE-based

hashing (Coleman & Shrivastava, 2020) to approximate the powered angular kernel. More importantly, YOSO lacks formal theoretical guarantees on its approximation quality and a mechanism that enables causal language modeling. RACE addresses both limitations: Sections 3.2 and 3.3 develop a theoretically grounded estimator with quantifiable approximation error, and Algorithm 2 presents a practical attention mechanism that naturally supports causal settings. We defer an in depth comparison between RACE and YOSO to Section 3.1. Two notable lines of work in the direction of linearly approximating attention are Linear Attention Katharopoulos et al. (2020) and Performer Choromanski et al. (2021). Linear Attention replaces the softmax similarity with a simple positive kernel via a feature map, e.g. $\phi(x) = elu(x) + 1$. This lets the attention be re-ordered into associative sums, achieving linear computation. Although such a kernel trick reduces computational complexity, it often degrades accuracy, as clearly demonstrated by our experiments in Section 4.1. Performer takes a different approach and cleverly leverages the classical idea of approximating the exponential of an inner product using Random Fourier Features Rahimi & Recht (2007). However, this strategy comes with its own drawbacks. In particular, the method incurs a time complexity that is quadratic in the embedding dimensionality, which offsets many of the intended computational savings. Furthermore, it is well established Backurs et al. (2017) that approximations based on Random Fourier Features require high-dimensional representations to achieve satisfactory accuracy. Our experimental results in Section 4.2 reinforce this limitation by showing that these methods exhibit poor scalability in practice.

Another category of work replaces the full $N \times N$ attention matrix with a low-rank surrogate. Some methods learn length-wise projections for keys/values (e.g., *Linformer*), while others use Nyström landmarks to approximate Softmax Attention matrix with a rank-k decomposition (e.g., *Nyströmformer*). These approaches reduce the leading cost from $O(N^2d)$ to $O(Nkd)$, at the cost of choosing (and occasionally increasing) k to maintain accuracy. (Wang et al., 2020; Xiong et al., 2021). Moreover, these methods provide no support for autoregressive tasks. As shown in Section 4.1, our method outperforms Linformer in accuracy despite Linformer having 13% more parameters than the other methods. Beyond the empirical shortcomings, a deeper conceptual issue persists: existing approximation approaches lack a rigorous mathematical framework to characterize the trade-offs between efficiency and accuracy. For example, while Performer provides strong kernel-approximation guarantees, a general framework connecting efficiency knobs (e.g., feature count m) to downstream accuracy remains limited, and strong accuracy frequently entails large m in practice. As a result, design decisions often appear ad hoc and fragile, leaving methods vulnerable to instability between tasks and settings. Taken together, these limitations explain why, despite the abundance of approximations, Softmax Attention continues to remain the most widely adopted and reliable formulation.

Sparsity is Complementary: We note that there is also a popular line of work (Beltagy et al., 2020; Zaheer et al., 2020; Kitaev et al., 2020; Han et al., 2023; Petrick et al., 2022) that exploits structural information in natural language, with sparsity in attention being among the most widely studied. These approaches are complementary to our proposal, which focus on making the fundamental attention mechanism itself more efficient and mathematically grounded. In principle, our method can be integrated with structural priors such as sparsity to further improve scalability and accuracy. However, since our objective in this paper is to develop fundamentally efficient attention, we will not discuss this line of structural approaches further, instead we view combining them with our method as an important direction for future work.

Our Finding: Standard attention relies on the well-known softmax function, computing

$$O = \mathrm{softmax}\Big(\tfrac{QK^\top}{\sqrt{d}}\Big)\, V\,, \tag{1}$$

where the softmax is applied row-wise so that attention weights are nonnegative and sum to one. In this paper, we propose a surprising alternative to the softmax—namely, a higher-degree monomial of an *angular* kernel based on cosine geometry:

$$O = \left(1 - \left(\frac{\cos^{-1}(QK^\top)}{\pi}\right)\right)^{\gamma} V \tag{2}$$

Eq. 2 should be read as an *informal* analogue to Eq. 1, where the angular kernel replaces the exponential. A more precise definition, with explicit cosine normalization and row-wise normalization, is given in Section 3.1. We argue that, for sufficiently large values of γ, this formulation closely

mimics the behavior of softmax and refer to it as *Angular Attention* in the subsequent sections. Importantly, it admits a linear-time approximation algorithm. In particular, we leverage the connection (Section 2.2) between **R**epeated **A**rrays-of-**C**ount **E**stimators (RACE) (Coleman & Shrivastava, 2020; Luo & Shrivastava, 2018) and the angular kernel to design our algorithm in Section 3.2. We therefore refer to our proposed method as *RACE Attention*.

RACE Attention is a drop-in replacement for Softmax Attention. We evaluate it in Transformers on causal language modeling, masked language modeling, and text/image classification (Section 4.1). By reframing similarity around a powered angular kernel and using differentiable LSH-based sketches, it provides a principled alternative that supports very long contexts on commodity hardware. The sketching view keeps constant factors small: each query mixes with only a fixed bank of $S = LR$ bucket summaries rather than all N keys. Since we never materialize the full attention matrix, the working set stays compact and activation memory drops, enabling much longer sequences with reduced latency. In contrast, FlashAttention-2/3 (Dao, 2023; Shah et al., 2024) reduce the memory footprint of attention via tiling, but still require computing all key-query interactions, preventing processing of much longer sequences at comparable speed, as shown in Section 4.2. In addition to our novel findings about RACE Attention and rigorous supporting experimental evidence, we provide the following:

- **Long-context scaling:** We show that RACE scales to context lengths well beyond those reported for prior attention mechanisms, handling up to *75M tokens* on a CPU and *12M tokens* on a GPU under the same hyperparameter settings used in our accuracy evaluations.
- **Trainable RACE:** We obtain a differentiable sketch by replacing hard hashing with smooth soft assignments over hypercube corners, thereby enabling end-to-end training.
- **CPU/GPU pre-training:** We support both *causal* (autoregressive) and *non-causal* (bidirectional) pre-training on CPU and GPU. For CPU workloads, we provide a custom OpenMP kernel that computes the causal prefix operations in a single pass using efficient streaming algorithm, enabling linear-time, memory-efficient training.
- **Theory and ablations:** Section 3.3 establishes approximation guarantees based on the LSH framework. We also characterize how the parameters L (number of hash tables) and R (buckets per table) influence the variance–accuracy tradeoff. Extensive ablations then measure accuracy, runtime, and memory as these parameters are varied.

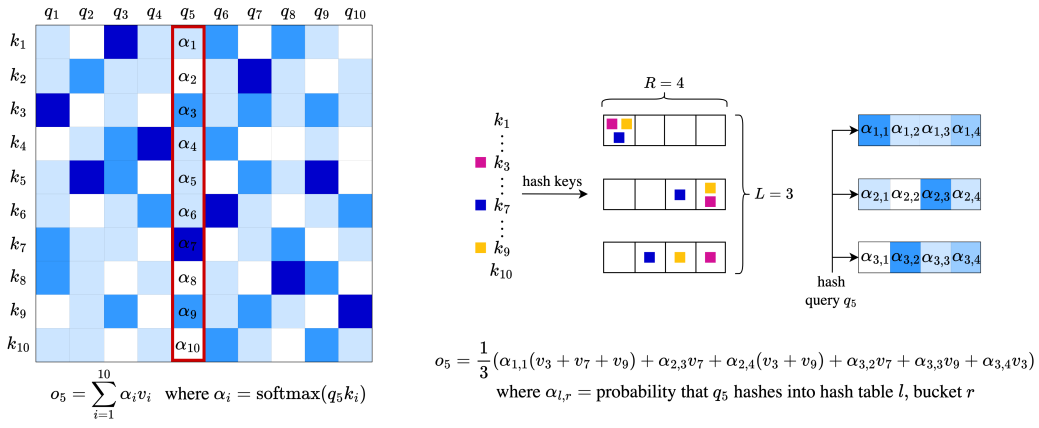


Figure 1: This figure demonstrates the difference between the linear complexity of RACE Attention and the quadratic complexity of Softmax Attention mechanism. Specifically, we highlight how the final representation o_5 is computed under Softmax versus RACE. In Softmax, the entire fifth column of the attention score matrix is required. In contrast, RACE does not require the full matrix; instead, it aggregates statistics within LSH-mapped buckets, utilizing the appropriate collision probability α to compute o_5.

2 BACKGROUND

2.1 LOCALITY-SENSITIVE HASHING (LSH)

An LSH family $\mathcal{H}$ for a similarity Sim makes near pairs collide more often than far pairs. Formally, $\mathcal{H}$ is (S_0, cS_0, p_1, p_2)-sensitive if for all $x, y \in \mathbb{R}^D$,

$$\begin{cases} \mathrm{Sim}(x,y) \geq S_0 \;\Rightarrow\; \Pr_{h\sim\mathcal{H}}[h(x)=h(y)] \geq p_1, \\ \mathrm{Sim}(x,y) \leq cS_0 \;\Rightarrow\; \Pr_{h\sim\mathcal{H}}[h(x)=h(y)] \leq p_2, \end{cases} \qquad p_1 > p_2,\ c < 1.$$

Such families enable sublinear-time approximate nearest-neighbor data structures. A convenient sufficient condition, satisfied by SimHash and WTA (Charikar, 2002; Yagnik et al., 2011; Chen & Shrivastava, 2018), is that the collision probability is a monotone function of similarity, $\Pr_{h\sim\mathcal{H}}[h(x) = h(y)] = f(\mathrm{Sim}(x,y))$ with f increasing.

2.2 RACE SKETCH

RACE (Coleman & Shrivastava, 2020; Coleman et al., 2020) shows that any similarity expressible as a (non-negative) linear combination of LSH collision kernels can be sketched using ACE-style estimation Luo & Shrivastava (2018). It provides an *unbiased* estimator of kernel-density sums for LSH collision kernels and their *powers*. In particular, RACE estimates $\sum_{x\in D} k(x,q)^p$ by hashing items into counters and reading the counters addressed by the query; averaging across L independent rows reduces variance.

Lemma 1 (Theorem 1 of (Coleman & Shrivastava, 2020))**.** *Given a dataset D, an LSH family H with finite range $[1, R]$ and a parameter p, construct an LSH function $h(x) \to [1, R^p]$ by concatenating p independent hashes from H. Let A be an ACE array constructed using $h(x)$. Then for any query q,*

$$\mathbb{E}\big[A[h(q)]\big] \;=\; \sum_{x\in D} k(x,q)^p$$

3 INTRODUCING RACE ATTENTION

3.1 SOFTMAX-LIKE SIMILARITIES THAT ADMIT LINEAR-TIME ESTIMATION

Given a sequence of N tokens, a Transformer produces for each position i a query $Q_i \in \mathbb{R}^d$, and for every position j a key $K_j \in \mathbb{R}^d$ and a value $V_j \in \mathbb{R}^d$. The output at position i is a weighted sum of the values, where the weight on V_j reflects the relevance of token j to token i. In the standard formulation (Vaswani et al., 2017), relevance is computed via the scaled dot product given by Eq. 1. This choice guarantees two useful properties of the attention weights: (i) non-negativity and (ii) they sum to one, so O_i is a convex combination of the values. Equally important, the exponential introduces a strong non-linear mapping from similarity scores to attention weights, amplifying small score differences. This observation suggests a broader view: attention weights can be derived from any normalized highly non-linear (exponential like) similarity function. Let $\mathrm{sim} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_{\geq 0}$ be any non-negative similarity function. We can define normalized *similarity attention* as

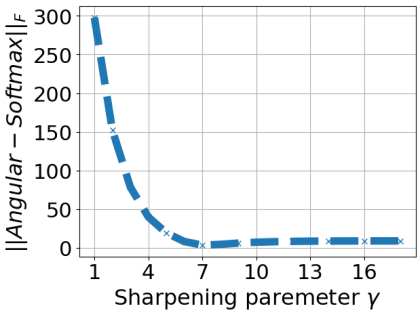


Figure 2: Frobenius error between Angular and Softmax Attention vs. γ.

$$O_i \;=\; \frac{\sum_{j=1}^{N} \mathrm{sim}(Q_i, K_j)\, V_j}{\sum_{j=1}^{N} \mathrm{sim}(Q_i, K_j)} \tag{3}$$

Our quest is for finding non-linear (softmax-like) similarity kernels that admit accurate linear-time estimation, eliminating the quadratic cost of attention in both training and inference. We argue that a good starting point is a well known LSHable (Coleman & Shrivastava, 2020; Choromanski et al., 2017) *angular* similarity. It is well behaved and normalized, in particular, it depends only on the angle between the vectors Q_i and K_j and is invariant to their norms: $\mathrm{sim}(Q_i, K_j) = 1 - \cos^{-1}\left(\frac{Q_i^\top K_j}{\|Q_i\|\,\|K_j\|}\right)/\pi$. However, unlike exponential in softmax, the raw angular kernel is relatively flat near high similarity values, reducing its ability to sharply discriminate between nearly aligned vectors. To increase contrast, we propose to exponentiate the angular kernel with a sharpening

parameter γ, which accentuates differences among highly similar pairs. After sharpening the kernel, the similarity function becomes as follows:

$$\mathrm{sim}(Q_i, K_j) = \left(1 - \cos^{-1}\left(\frac{Q_i^\top K_j}{\|Q_i\|\,\|K_j\|}\right)/\pi\right)^\gamma \tag{4}$$

In fig. 3, we show that for sufficiently large γ, the angular kernel becomes almost indistinguishable from softmax kernel. This is expected because a higher degree monomial like x^{12} behaves similarly to an exponential. Furthermore, in fig. 2 we plot the frobenius error between Angular and Softmax Attention. The error sharply decreases as γ increases, demonstrating that softmax-level sharpness can be achieved with modest polynomial degree (*e.g.*, $\gamma = 8$).

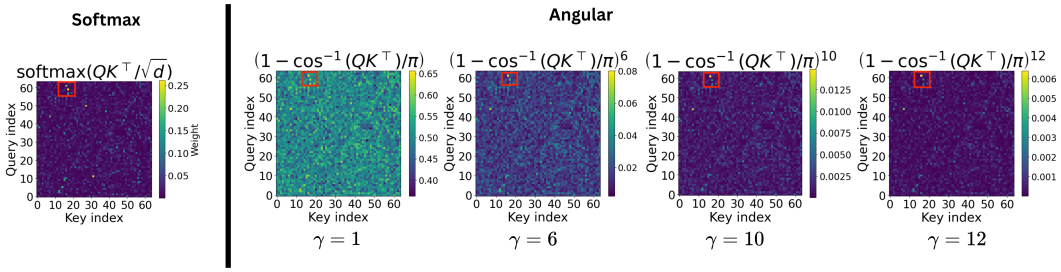


Figure 3: Comparison of Softmax and Angular kernels at different sharpening levels γ. As γ (or non-linearity) increases, Angular transitions from flat similarity scores to a sharper distribution, recovering behavior similar to the exponential in the Softmax.

Algorithm 1 RACE Attention (non-causal)

Input: $Q, K, V \in \mathbb{R}^{N\times d}$; number of hash tables L; number of hyperplanes P; temperature $\beta > 0$.
Output: $\widehat{O} \in \mathbb{R}^{N\times d}$.

1: **for** $\ell = 1, \ldots, L$ **do**
2: Draw $W^{(\ell)} \in \mathbb{R}^{P\times d}$ with rows $w_p^{(\ell)} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_d)$.
3: Define the corner set $\mathcal{V} = \{\pm 1\}^P$ ($R = 2^P$) with $v_r \in \{\pm 1\}^P$.
4: Build $\Phi_Q^{(\ell)}, \Phi_K^{(\ell)} \in \mathbb{R}^{N\times R}$ with rows

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

5: Per-table bucket statistics:

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

6: **end for**
7: Compute average across tables: $\texttt{Num} = \frac{1}{L}\sum_{\ell=1}^L \Phi_Q^{(\ell)} B^{(\ell)}$ and $\texttt{Den} = \frac{1}{L}\sum_{\ell=1}^L \Phi_Q^{(\ell)} A^{(\ell)}$.
8: Return $\widehat{O} \leftarrow \mathrm{diag}(\texttt{Den})^{-1}\,\texttt{Num}$

In its current form, evaluating the attention with similarity given by Eq. 4 is no different from softmax. It naively still requires all N^2 query-key interactions. Fortunately, any constant exponentiation of angular kernel, belongs to a family, that admits efficient kernel density estimation using RACE sketches Coleman & Shrivastava (2020), and we use these sketches to approximate the kernel in linear time obtaining an algorithmically efficient alternative to Softmax Attention!

Finally, as outlined in Section 1, YOSO Zeng et al. (2021) and our method operate on the same kernel as in Eq. 4, but the underlying attention computation differs significantly. YOSO uses hard LSH hashing to construct Bernoulli collision indicators, which provide an unbiased estimator of the unnormalized numerator $\sum_{j=1}^N \mathrm{sim}(Q_i, K_j)V_j$ in Eq. 3. However, YOSO does not estimate the normalization term $\sum_{j=1}^N \mathrm{sim}(Q_i, K_j)$ in Eq. 3, instead, it applies a post-hoc ℓ_2 normalization strategy, which does not correspond to the standard attention normalization. Moreover, since YOSO's hard LSH is non-differentiable, it uses surrogate lower-bound gradients derived from additional Bernoulli

samples instead of differentiating the kernel directly, leading to a quadratic time dependence on the embedding dimension d during end-to-end training. To see how RACE constructs a smooth, differentiable LSH-based estimator that is linear in d in contrast to YOSO, refer to Section 3.2.

3.2 The Final Algorithm

At a high level, RACE does not approximate the $N \times N$ score matrix (which would remain quadratic). Instead, it sketches the sufficient statistics needed to compute the attention outputs directly, yielding a linear-time approximation. Fig. 1 illustrates this distinction by focusing on o_5, the output embedding of token 5 after attention. In Softmax Attention, computing o_5 requires the entire column of the attention matrix to get the weighted combination of vectors. RACE Attention, in contrast, employs LSH-indexed hash functions to softly assign the N keys and values (Ks and Vs) into R representative *bucket summaries*. After this assignment, each query (the Q) is softly hashed into buckets under the same LSH scheme. For example, the output for token 5, o_5, is obtained by mixing the summaries of tokens assigned to the same bucket as the one mapped by Q_5 under LSH. The mixing is weighted by soft probability values derived from the well-defined collision probability of the LSH mapping function. Averaging across L independent tables, a standard sketching technique, further reduces variance and stabilizes the approximation. A complete step-by-step expansion is provided in Appendix (fig. 9) and Algorithm 1. Furthermore, we also provide an intuitive visualization of how similarities between tokens look like across buckets in Appendix (fig. 10).

We next formalize the RACE Attention mechanism in Algorithm 1. As described in Section 2.2, one final technical hurdle remains: the RACE algorithm is non-differentiable. We get around the non-differentiability of RACE sketches by replacing discrete bucket assignments with soft probabilities and using standard cross-entropy loss, preserving differentiability for end-to-end training. Algorithm 1 consists of three key stages: (i) **Soft bucketization:** Each query/key $x \in \mathbb{R}^d$ is randomly projected via $W^{(\ell)}$ hyperplanes and softly assigned to $R = 2^P$ corners with distribution $\phi^{(\ell)}(x)$ (steps 2–4), (ii) **Bucket aggregation:** For each table ℓ, we form per-bucket statistics by accumulating key weights and their weighted values, namely the mass vector $A^{(\ell)} \in \mathbb{R}^R$ and the value-sum matrix $B^{(\ell)} \in \mathbb{R}^{R\times d}$, so that $A^{(\ell)}[r]$ is the total (soft) mass in bucket r and $B^{(\ell)}[r,:]$ is the corresponding sum of values. (step 5), (iii) **Global normalization:** The algorithm averages across L tables to form $\texttt{Num} = \frac{1}{L}\sum_\ell \Phi_Q^{(\ell)} B^{(\ell)}$ and $\texttt{Den} = \frac{1}{L}\sum_\ell \Phi_Q^{(\ell)} A^{(\ell)}$, and reconstructs the final outputs as $\widehat{O} = \mathrm{diag}(\texttt{Den})^{-1}\,\texttt{Num}$ (steps 7–8).

A useful way to interpret Algorithm 1 is through the kernel perspective introduced in Section 2.2. In a classical RACE Coleman & Shrivastava (2020), P random hyperplanes generate a hash $h(x) = \mathrm{sign}(W^{(\ell)}x)$, and two vectors collide under this hash with probability $\Pr[h(Q_i) = h(K_j)] = S_{ij} := \mathrm{sim}(Q_i, K_j)$, which is exactly the P-powered angular kernel in Eq. 4 with $\gamma = P$. In soft RACE, we keep this geometric structure intact, but replace the hard sign map with a smooth approximation: we compute a "soft" sign vector $\tanh(W^{(\ell)}x)$ and evaluate its alignment with each of the $R = 2^P$ corner sign patterns $v_r \in \{\pm 1\}^P$, assigning x to buckets with nonzero probability via a softmax over these alignments. This turns the discrete collision event of classical RACE into a differentiable quantity while preserving its underlying angular dependence. In particular, vectors with small angular distance still assign most of their mass to the same buckets, mirroring the behavior of the P-powered angular kernel. As a result, the per-table quantity $\phi^{(\ell)}(Q_i)^\top \phi^{(\ell)}(K_j)$ serves as a smooth approximation to the P-powered angular similarity that our attention mechanism seeks to approximate. We formalize these kernel quantities in the Section 3.3, where the RACE-based approximation $\widehat{S}_{ij}$ is introduced.

Computational Complexity: The per-table runtime of Algorithm 1 can be decomposed according to its main steps: Step 2 (random projections) costs $\mathcal{O}(NdP)$, Step 3 (logits over $R = 2^P$ corners) costs $\mathcal{O}(NPR)$, and Step 5 (bucket aggregation) costs $\mathcal{O}(NRd)$. The global accumulation in Step 7 adds $\mathcal{O}(NRd)$ *per table*. Thus, the per-table runtime is $\mathcal{O}(NdP + NPR + NRd) = \mathcal{O}(NRd)$, with memory $\mathcal{O}(NR + Rd)$. Across L tables, this becomes $\mathcal{O}(LNRd)$ time and $\mathcal{O}(L(NR + Rd))$ space. Compared to Softmax Attention's (FlashAttention-2/3) $\mathcal{O}(N^2 d)$ time and $\mathcal{O}(Nd)$ space, RACE is more efficient since $R, L \ll N$ and $R, L \ll d$, even for moderate N and d.

3.3 Theoretical Analysis of Algorithm 1

Algorithm 1 is presented in terms of random projections, soft bucketization, and per-bucket aggregation. To analyze its quality, we now make the kernel viewpoint introduced above precise. Each hash table $\ell = 1, \ldots, L$ induces a randomized feature map $\phi^{(\ell)} : \mathbb{R}^d \to \mathbb{R}^R$, where $R = 2^P$ is

the number of hypercube corners, and defines the approximate kernel $\widehat{S}_{ij}^{(\ell)} = \left(\phi^{(\ell)}(Q_i)\right)^\top \phi^{(\ell)}(K_j)$. Then, averaging across L independent tables yields $\widehat{S} = \frac{1}{L}\sum_{\ell=1}^{L} \widehat{S}^{(\ell)}$. This view places Soft RACE Attention in the language of kernel methods: it replaces the angular kernel (with $\gamma = P$) in Eq. 4 with the randomized sketch $\widehat{S}$ based on LSH-style features. Since $\phi^{(\ell)}(x)$ is a softmax distribution, the approximate kernel $\widehat{S}$ inherits concentration properties from the underlying random Gaussian projections. This allows us to analyze its deviation from the target angular kernel using standard tools from randomized numerical linear algebra (RandNLA). Our analysis requires the following two mild assumptions on the target kernel S:

(A1) Row sums of S are bounded away from zero *i.e.,* $s_{\min} := \min_i (S\mathbf{1})_i \geq C_1 N$ for some constant $C_1 > 0$, which ensures stable normalization in attention.
(A2) Spectral norm of S is bounded *i.e.,* $\|S\|_2 \leq C_2 N$, which follows from $S_{ij} \in [0, 1]$.

Several comments are necessary to better understand the above structural conditions. Condition (A1) rules out degenerate cases where a query has vanishing similarity with all keys, which would make the row-normalization in attention unstable. In practice this is mild: each row of S naturally carries a nontrivial fraction of mass (as embeddings are trained to maintain similarity), so requiring $s_{\min} \geq cN$ is only a safeguard against pathological isolation of a query. Condition (A2) is even less restrictive: since $S_{ij} \in [0, 1]$, the worst case is the all-ones matrix J_N, which has spectral norm exactly $\|J_N\|_2 = N$. Thus bounding $\|S\|_2 \leq C_2 N$ merely rules out pathological growth beyond this trivial maximum, and is always satisfied up to a constant factor.

We are now ready to state our main quality-of-approximation result:

Theorem 2. *Let $Q, K, V \in \mathbb{R}^{N\times d}$ be the query, key, and value matrices. For parameters L, P, and β, and under conditions (A1) and (A2), the estimator $\widehat{O}$ produced by Algorithm 1 satisfies*

$$\|\widehat{O} - O\|_F \;=\; \mathcal{O}\left(\left(\tfrac{P}{\beta} + \sqrt{\tfrac{\log(N/\delta)}{L}}\right)\|V\|_F\right)$$

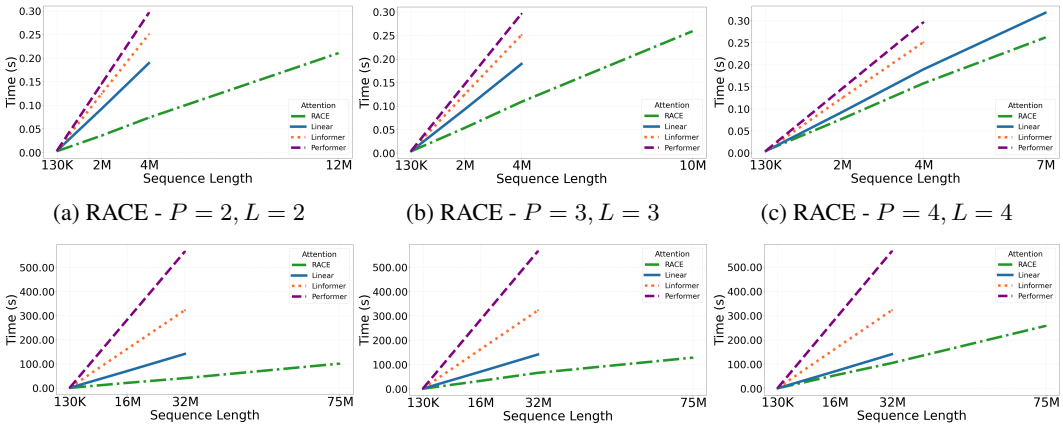
with probability at least $1 - \delta$. Recall that $O \in \mathbb{R}^{N\times d}$ with the i^{th} row O_i is defined using Eqs. 3 and 4 with $\gamma = P$.

The bound decomposes into a *bias term* $\mathcal{O}(P/\beta)$ and a *variance term* $\mathcal{O}(\sqrt{\log(N/\delta)/L})$. Larger β reduces the bias, while increasing L reduces the variance. The dependence on P arises because powering the angular kernel by P makes collisions sharper, but soft bucketization (finite β) smooths out these decisions and introduces additional bias. To keep this bias small, β should be scaled with P. In particular, as $\beta, L \to \infty$, the approximation error vanishes. In fact taking $L = \Theta(\log N)$ prevents the variance from exploding. Together, this kernel reinterpretation provides a precise RandNLA lens for analyzing *RACE Attention*, with L, P, and β jointly governing its accuracy-efficiency trade-offs. The proof of Theorem 2, together with all intermediate lemmas, is deferred to Appendix C due to space constraints.

Remark (Causal masking): Our language modeling experiments in Section 4.1 employ RACE Attention with causal masking, implemented efficiently via OpenMP for CPU workloads. See Algorithm 2 in the Appendix for the causal soft RACE Attention algorithm. Our theoretical analysis above applies only to the non-causal setting. Extending the bias–variance guarantees of Theorem 2 remains an open problem, as the cumulative-sum constraint interacts non-trivially with the random feature construction. A rigorous analysis of causal RACE Attention is an important direction for future work.

4 Experiments

To avoid cherry-picking and ensure comparability, we adopt the evaluation suites standard in prior efficient-attention work—Linear Attention, Linformer, and Random Feature Attention (RFA) (Katharopoulos et al., 2020; Wang et al., 2020; Peng et al., 2021). Concretely, we include **text classification** (QNLI Rajpurkar et al. (2016), SST-2 Socher et al. (2013), IMDB Maas et al. (2011), Yahoo Zhang et al. (2015)), and Arxiv He et al. (2019) to probe moderate-length and long-context discriminative accuracy (as in Linformer); **autoregressive language modeling** to test token-level modeling (as in RFA) on WikiText-103 Merity et al. (2017) and Penn Tree Bank (PTB) corpus Marcus et al. (1993); **masked language modeling** on Tiny Stories dataset Eldan & Li (2023); **image classification** (CIFAR-10 Krizhevsky (2009), FashionMNIST Xiao et al. (2017), Food-101 Bossard et al. (2014)) to test expressivity using Vision Transformer Dosovitskiy et al. (2021) architecture



(a) RACE - $P = 2, L = 2$ (b) RACE - $P = 3, L = 3$ (c) RACE - $P = 4, L = 4$

(d) RACE - $P = 2, L = 2$ (e) RACE - $P = 3, L = 3$ (f) RACE - $P = 4, L = 4$

Figure 4: A rigorous scaling stress-test across hardware. The top row shows GPU scaling results; the bottom row shows CPU scaling results. All plots use logarithmic axes. We run a single forward-backward pass configured with 1 batch, 4 heads, and an embedding dimension of 128. Linformer and Performer use the same low-rank/feature dimension as in Table 1.

(as in Linear Attention); and **long-context reasoning** via Long Range Arena benchmark Tay et al. (2021) (e.g., ListOps and Text Retrieval) to stress scaling and accuracy. Together these cover four regimes—bidirectional, autoregressive, long-context, and moderate-context text/image classification. Beyond these standard benchmarks, we also conduct extreme-length scaling experiments, reaching sequence lengths in the tens of millions of tokens, using the same hyperparameter configuration as in our accuracy evaluations. We additionally define M, the number of ensembles per head (independent sketch replicas). Since all experiments use $M = 1$, we omit this hyperparameter from all tables/figures for simplicity. To the best of our knowledge, this is the first work to report experiments with attention spanning close to 100 million tokens. For clarity, FlashAttention-2/3 are exact, fused-kernel implementations of Softmax Attention; we therefore use "FlashAttention-2/3" and "Softmax Attention" interchangeably when discussing accuracy and runtime.

Baselines: We compare RACE against widely used baselines with publicly available implementations: FlashAttention2 Dao et al. (2022), Linear Attention Katharopoulos et al. (2020), Performer Choromanski et al. (2021), Linformer Wang et al. (2020), and Sigmoid Attention Ramapuram et al. (2025). These span exact, kernel-linear, and low-rank approximations. All models are tuned per authors' guidelines and trained under identical settings.

4.1 Is RACE Attention as Accurate as Transformers?

We report **text-classification** accuracies in Tables 4, 7, 13, 14, and 15; **long-context (LRA)** results (ListOps and Text Retrieval) in Tables 3 and 5; **image-classification** accuracies in Tables 1 and 10; and **masked language modeling (MLM)** perplexities in Tables 2 and 11. For **autoregressive language modeling**, RACE matches softmax-level perplexity on WikiText-103 and improves upon it on PTB (Tables 6 and 12). These results indicate that RACE preserves accuracy in the overlapping regime while delivering consistent gains on moderate-context and long-context settings. Unless stated otherwise, all methods use the same Transformer backbone (layers, heads, embedding dimension, dropout) and training budget. We train with identical optimizers, schedulers, and batch sizes; full hyperparameters appear in Table 8. Metrics are reported from the best-validation checkpoint. Despite the extra parameters introduced by Linformer's lengthwise projections, it does not outperform RACE under matching training conditions. All experiments were run on an NVIDIA A100 GPU.

4.2 Can we reach 100 million context window on popular hardware?

In this section, we evaluate how RACE Attention scales across common hardware relative to strong baselines. For RACE, we use sketch parameters (P, L) chosen to match FlashAttention2's accuracy/perplexity on the same tasks in Section 4.1. For each method, we measure the wall-clock time for a single forward–backward pass of the multi-head attention layer with 1 batch, 4 heads, and embedding dimension of 128, as a function of sequence length, stress-testing context lengths up to 100 million tokens. Since FlashAttention-2/3 are designed specifically for GPU hardware, we use PyTorch's optimized "F.scaled_dot_product_attention" implementation as the FlashAttention base-

Table 1: **Sequential CIFAR-10 @ N=1024**

Method	Accuracy
RACE (P=2, L=2)	63.7%
RACE (P=3, L=3)	62.5%
RACE (P=3, L=5)	65.7%
RACE (P=4, L=5)	**65.9%**
Linformer-128	63.7%
FlashAttention2	61.44%
Angular (γ=8)	61.69%
Linear	60.0%
Performer-256	64.9%
Sigmoid	57.2%

Table 2: **Tiny Stories (subset) @ N=1024**

Method	Perplexity
RACE (P=2, L=2)	4.2
RACE (P=3, L=3)	3.2
RACE (P=4, L=4)	2.6
Linear	7.0
Linformer-128	3.7
FlashAttention2	2.7
Angular (γ=8)	**2.5**
Performer-256	10.0
Sigmoid	3.7

Table 3: **Text Retrieval @ N=8000**

Method	Acc.
RACE (P=2, L=2)	80.3%
RACE (P=2, L=3)	80.5%
RACE (P=3, L=3)	80.8%
RACE (P=4, L=4)	**80.9%**
Linformer-128	76.1%
Linear	80.6%
Performer-256	80.8%
FlashAttention2	80.5%

Table 4: **QNLI @ N=2048**

Method	Accuracy
RACE (P=2, L=2)	60.7%
RACE (P=3, L=3)	60.7%
RACE (P=4, L=4)	61.1%
RACE (P=5, L=5)	60.4%
Linformer-128	60.6%
Linear	60.7%
FlashAttention2	61.1%
Angular (γ=8)	**61.7%**
Performer-256	61.0%
Sigmoid	61.1%

Table 5: **ListOps @ N=2000**

Method	Acc.
RACE (P=2, L=2)	41.9%
RACE (P=2, L=3)	41.0%
RACE (P=3, L=3)	41.3%
RACE (P=4, L=3)	41.2%
Linformer-128	38.9%
FlashAttention2	41.4%
Angular (γ=8)	**42.2%**
Linear	39.6%
Performer-256	40.2%

Table 6: **Wiki-103 @ N=1024**

Method	Test PPL
RACE (P=2, L=2)	23.9
RACE (P=2, L=3)	23.4
RACE (P=3, L=3)	21.9
RACE (P=3, L=4)	21.5
RACE (P=4, L=4)	20.9
FlashAttention2	20.9
Angular (γ=8)	**19.0**

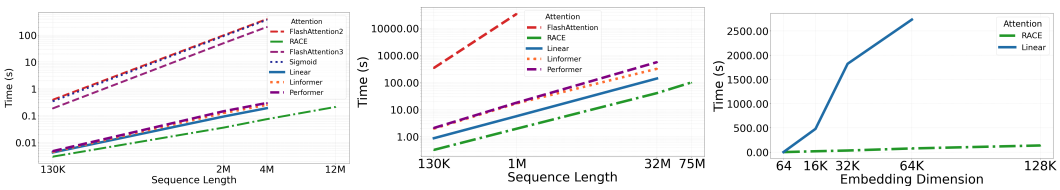
Table 7: Long-context classification performance on a 40GB A100 GPU. Train/Test denote per-epoch runtimes in seconds, and Acc. denotes accuracy.

Arxiv Long-Document Classification									
	16K			**32K**			**64K**		
Method	**Train** $\downarrow$	**Test** $\downarrow$	**Acc.** $\uparrow$	**Train** $\downarrow$	**Test** $\downarrow$	**Acc.** $\uparrow$	**Train** $\downarrow$	**Test** $\downarrow$	**Acc.** $\uparrow$
RACE (P=2,L=2)	**80.5s**	3.9s	70.3%	**282s**	15s	89.4%	**561s**	22s	97.14%
RACE (P=3,L=3)	82.4s	4.0s	**71.3%**	289s	15.6s	90.6%	584s	22.5s	97.92%
RACE (P=4,L=4)	84.7s	4.1s	70.8%	300s	16s	**91.1%**	594s	22.9s	97.4%
Linear	83.8s	4.0s	67.9%	286s	15.9s	87.3%	591s	22.8s	96.35%
Linformer-128	86s	**3.2s**	64.1%	296s	**10.7s**	87.5%	616s	**15.2s**	97.4%
Performer-256	128s	5.8s	68.9%	449s	24.6s	86.5%	952s	35s	96.61%
FlashAttention2	95.7s	3.7s	69.8%	471s	20s	89.7%	1645s	47s	97%

line when reporting CPU scaling results.

How far can we scale attention on standard Intel Xeon® Gold 5220R CPU?

RACE scales to **75 million** tokens for a single forward-backward pass on CPU. By contrast, FlashAttention becomes prohibitively slow at ~ 2 million tokens due to the quadratic time scaling in sequence length N (see fig. 5). It is worth noting that FlashAttention does not run out of memory on the CPU DRAM. RACE is more than $10000\times$ faster than FlashAttention at context length of ~ 33 million. RACE finishes comfortably under 10 seconds for a single forward-backward pass on this hardware while FlashAttention takes approximately 10^5 seconds on the same hardware. Although the absolute runtime grows with N, RACE's speedup over FlashAttention increases. At 75 million tokens, RACE finishes in about 100 seconds. This is expected because RACE is linear and FlashAttention is quadratic in N. The experiments also highlight that linear attentions' approximations are not only inaccurate but also significantly slower and have large memory overheads due to large hid-



(a) GPU scaling (b) CPU scaling (c) d scaling @ N=65536

Figure 5: A rigorous scaling stress-test across hardware. Plots (a)–(b) use logarithmic axes. RACE is evaluated with $(P{=}2, L{=}2)$ throughout; Linformer and Performer use the same low-rank/feature dimension as in Table 1.

den constants (see fig. 4). They run about an order of magnitude slower than RACE Attention and even go out of memory at $\sim$33 million tokens.

How far can we scale attention on the most powerful GH200 GPU?
An NVIDIA GH200 has 96GB of memory. Here, we observe a similar trend. RACE scales up to **12 million** tokens for a single forward-backward pass, whereas FlashAttention-2/3 and Sigmoid Attention becomes impractical around ~ 4 million tokens (see fig. 5). At ~ 4 million tokens RACE takes merely 0.1 seconds to finish, while FlashAttention2 needs about 550 seconds, making RACE about $5500\times$ faster on GPUs for processing 4 million tokens. Additionally, RACE is about $5000\times$ faster than Sigmoid and $2600\times$ faster than FlashAttention3 for processing 4 million tokens (see fig. 7). While FlashAttention's *activation memory* scales linearly with N and d, the GPU's high-bandwidth memory (HBM) is nevertheless exhausted for sufficiently large N. This is because we must retain Q, K, V, O (and their gradients) of size $\mathcal{O}(BHNd)$ with large constants. Even though the $N \times N$ score matrix is never materialized, this footprint exceeds HBM capacity at large N, leading to out-of-memory failures. Furthermore, RACE even scales better on GPU than the cheap but less accurate linear baselines, and they run out of memory around ~ 4 million tokens (see fig. 4). RACE handles about $3.5\times$ longer contexts than FlashAttention-2/3 and Sigmoid Attention.

4.3 Right Algorithm Beats Hardware Acceleration!

While GPUs offer substantial speedups for a fixed algorithm, comparing FlashAttention-2/3 and Sigmoid Attention on a high-end GH200 GPU against RACE Attention on a single CPU highlights the impact of *algorithmic* acceleration. Fig. 6 reports the runtime of a single forward–backward pass as the context length increases. For short to moderate sequences ($N \lesssim 131$K), the GPU's massive parallelism dominates and FlashAttention-2/3 remain faster. Beyond this scale, however, the asymptotic behavior becomes decisive: the GPU kernels begin to saturate, and RACE becomes faster. At a sequence length of ~ 4 million tokens (the largest supported by FlashAttention-2/3 in our configuration), RACE on CPU is roughly $40\times$ faster than FlashAttention-2 and Sigmoid Attention on GPU, and about $20\times$ faster than FlashAttention-3 (see fig. 8). These results show that, in the long-context regime, state-of-the-art GPU attention kernels are fundamentally constrained by their quadratic dependence on sequence length. In contrast, RACE Attention's algorithmic efficiency enables it to outperform even the most advanced GPU-accelerated methods by a wide margin, despite executing on substantially weaker hardware.

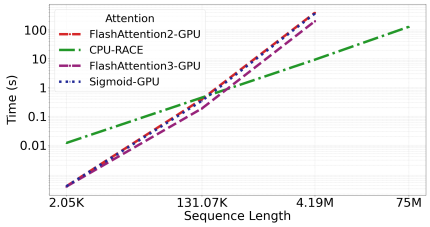


Figure 6: A rigorous scaling test for algorithmic comparison between FlashAttention-2/3 and Sigmoid Attention on GPU vs. RACE Attention ($P = 3, L = 3$) on CPU. This plot uses logarithmic axes.

5 Conclusion

We introduced RACE Attention, a linear-time, memory-efficient alternative to softmax attention that approximates a sharpened angular kernel using RACE sketches. By replacing pairwise scores with aggregated bucket statistics, RACE scales linearly in context length N and embedding dimension d, with constants set by sketch parameters (P, L) while providing good accuracy at long context-lengths. RACE Attention further opens opportunities for cache-free inference and CUDA kernels that extend these scaling properties to autoregressive workloads.

Ethics Statement

Our work is focused on speeding up algorithm and reducing the memory complexity for Attention. As such, it could have significant broader impacts by allowing practitioners to train models fast and deploy them in constrained resource settings. Our experimental work uses publicly available datasets to evaluate the performance of our algorithm; no ethical considerations are raised.

Reproducibility statement

We provide the source code and configuration for the key experiments (Language Modelling, Masked Language Modelling, and Classification) including instructions on how to generate data and train the models. All proofs are stated in the appendix with explanations and underlying assumptions. We thoroughly checked the implementation and also verified empirically that the proposed algorithm hold.

References

Arturs Backurs, Piotr Indyk, and Ludwig Schmidt. On the fine-grained complexity of empirical risk minimization: Kernel methods and neural networks. *arXiv preprint arXiv:1704.02958*, 2017.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

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

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014.

Moses S. Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC'02)*, pp. 380–388. ACM, 2002. doi: 10.1145/509907.509965.

Beidi Chen and Anshumali Shrivastava. Densified winner take all (wta) hashing for sparse datasets. In *Proceedings of the 34th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2018. arXiv:1810.00115.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamás Sarlós, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Łukasz Kaiser, David Belanger, Lucy Colwell, and Adrian Weller. Rethinking attention with performers. In *International Conference on Learning Representations*, 2021.

Krzysztof M. Choromanski, Mark Rowland, and Adrian Weller. The unreasonable effectiveness of structured random orthogonal embeddings. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems (NeurIPS) 30*. Curran Associates, Inc., 2017.

Benjamin Coleman and Anshumali Shrivastava. Sub-linear RACE sketches for approximate kernel density estimation on streaming data. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20–24, 2020*, pp. 1739–1749. ACM / IW3C2, 2020. doi: 10.1145/3366423.3380244.

Benjamin Coleman, Richard Baraniuk, and Anshumali Shrivastava. Sub-linear memory sketches for near neighbor search on streaming data. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 2089–2099. PMLR, 13–18 Jul 2020.

Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023. https://doi.org/10.48550/arXiv.2307.08691.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. Universal transformers. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, New Orleans, LA, USA, 2019. OpenReview.net.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

Ronen Eldan and Yuanzhi Li. Tinystories: How small can language models be and still speak coherent english? *arXiv preprint arXiv:2305.07759*, 2023.

Insu Han, Rajesh Jayaram, Amin Karbasi, Vahab Mirrokni, David P. Woodruff, and Amir Zandieh. Hyperattention: Long-context attention in near-linear time. *arXiv preprint arXiv:2310.05869*, 2023.

Jun He, Liqun Wang, Liu Liu, Jiao Feng, and Hao Wu. Long document classification from local word glimpses via recurrent attention learning. *IEEE Access*, 7:40707–40718, 2019. doi: 10.1109/ACCESS.2019.2907992.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *Proceedings of the 37th International Conference on Machine Learning*. PMLR, 2020.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *International Conference on Learning Representations*, 2020.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Chen Luo and Anshumali Shrivastava. Arrays of (locality-sensitive) count estimators (ace): Anomaly detection on the edge. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, pp. 1439–1448, New York, NY, USA, 2018. ACM. doi: 10.1145/3178876.3186056.

Haoneng Luo, Shiliang Zhang, Ming Lei, and Lei Xie. Simplified self-attention for transformer-based end-to-end speech recognition. *CoRR*, abs/2005.10463, 2020.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 142–150, 2011.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330, 1993.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4052–4061, Stockholm, Sweden, 2018. PMLR.

Hao Peng, Nikolaos Pappas, Dani Yogatama, Roy Schwartz, Noah A. Smith, and Lingpeng Kong. Random feature attention. In *International Conference on Learning Representations (ICLR)*, 2021.

Frithjof Petrick, Jan Rosendahl, Christian Herold, and Hermann Ney. Locality-sensitive hashing for long context neural machine translation. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pp. 32–42, Aachen, Germany, May 2022. Association for Computational Linguistics.

Zhen Qin, Weixuan Sun, Hui Deng, Dongxu Li, Yunshen Wei, Baohong Lv, Junjie Yan, Lingpeng Kong, and Yiran Zhong. cosFormer: Rethinking softmax in attention. *arXiv preprint arXiv:2202.08791*, 2022.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems 20 (NeurIPS 2007)*, pp. 1177–1184, Vancouver, British Columbia, Canada, 2007. Curran Associates, Inc.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2383–2392, 2016.

Jason Ramapuram, Federico Danieli, Eeshan Dhekane, Floris Weers, Dan Busbridge, Pierre Ablin, Tatiana Likhomanenko, Jagrit Digani, Zijin Gu, Amitis Shidani, and Russ Webb. Theory, analysis, and best practices for sigmoid self-attention. In *International Conference on Learning Representations (ICLR)*, 2025.

Jay Shah, Ganesh Bikshandi, Ying Zhang, Vijay Thakkar, Pradeep Ramani, and Tri Dao. Flashattention-3: Fast and accurate attention with asynchrony and low-precision. In *NeurIPS 2024*, 2024. arXiv preprint arXiv:2407.08608.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1631–1642, 2013.

Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. Long range arena: A benchmark for efficient transformers. In *International Conference on Learning Representations (ICLR)*, 2021.

Joel A. Tropp. An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230, 2015.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. Nyströmformer: A nyström-based algorithm for approximating self-attention. *arXiv preprint arXiv:2102.03902*, 2021.

Jay Yagnik, Dennis Strelow, David A. Ross, and Ruei-Sung Lin. The power of comparative reasoning. In *2011 International Conference on Computer Vision (ICCV)*, pp. 2431–2438. IEEE, 2011. doi: 10.1109/ICCV.2011.6126540.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Zhanpeng Zeng, Yunyang Xiong, Sathya Ravi, Shailesh Acharya, Glenn M. Fung, and Vikas Singh. You only sample (almost) once: Linear cost self-attention via bernoulli sampling. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12321–12332. PMLR, 2021.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

A Appendix

A.1 LLM Usage

We only use LLM to polish our language in the entire paper. No LLM is used for generating ideas or results.

B Additional Notes on Experiments

Table 8: Experiment Setup and Hyperparameters

Dataset	Task	Hyperparameters
PTB	Language Modeling	N=128; layers=1; heads=2; d=128; batch=16; lr=$6e^{-4}$; β's=(0.9, 0.999); ϵ=$1e^{-8}$; wd=0.1; dropout = 0.3; epochs=70
WikiText-103	Language Modeling	N=1024; layers=8; heads=8; d=512; batch=16; lr=$6e^{-4}$; β's=(0.9, 0.999); ϵ=$1e^{-8}$; wd=0.1; dropout=0.1; epochs=100
IMDB	Text Classification	N=512; layers=1; heads=2; d=128; batch=32; lr=$1e^{-5}$; wd=$5e^{-5}$; dropout=0.1; epochs=150
Yahoo	Text Classification	N=256; layers=1; heads=2; d=128; batch=32; lr=$1e^{-5}$; wd=$5e^{-5}$; dropout = 0.1; epochs=100
ListOps	Text Classification	N=2000; layers=8; heads=8; d=512; batch=16; lr=$1e^{-5}$; wd=$1e^{-5}$; dropout=0.1; epochs=40
Text Retrieval	Text Classification	N=8000; layers=4; heads=2; d=384; batch=1; lr=$2e^{-4}$; wd=$1e^{-2}$; dropout=0.1; epochs=20
Tiny Stories	Masked Language Modelling	N=512; layers=6; heads=4; d=384; batch=32; lr=$6e^{-4}$; β's=(0.9, 0.999); ϵ=$1e^{-8}$; wd=0.1; dropout=0.1; epochs=100; stories=20000
QNLI	Text Classification	N=2048; layers=4; heads=8; d=384; batch=32; lr=$1e^{-5}$; wd=$5e^{-5}$; dropout=0.1; epochs=100
SST-2	Text Classification	N=1024; layers=4; heads=8; d=384; batch=32; lr=$1e^{-5}$; wd=$5e^{-5}$; dropout=0.1; epochs=100
CIFAR-10	Image Classification	N=1024; layers=2; heads=4; d=384; batch=32; lr=$6e^{-4}$; β's=(0.9, 0.999); ϵ=$1e^{-8}$; wd=0.1; dropout=0.1; epochs=75
FashionMNIST	Image Classification	N=784; layers=2; heads=4; d=384; batch=32; lr=$6e^{-4}$; β's=(0.9, 0.999); ϵ=$1e^{-8}$; wd=0.1; dropout=0.1; epochs=75
Food-101	Image Classification	N=16384; layers=8; heads=8; d=512; batch=8; lr=$3e^{-4}$; wd=0.001; dropout=0.1; epochs=100; grad_accum_steps=4
Arxiv	Text Classification	N=16K-64K; layers=4; heads=4; d=256; batch=2; lr=$3e^{-4}$; wd=0.01; dropout=0.1; epochs=100; grad_accum_steps=16

Description: Unless otherwise stated, we use a linear warmup–decay learning-rate schedule. Let T denote the total number of optimizer updates ($T = \text{epochs} \times \text{len}(\text{train\_loader})$). The learning rate increases linearly from 0 to the base value (provided in Table 8) over the first $0.01T$ updates, then decays linearly to 0 over the remaining $T - 0.01T$ updates; the scheduler is stepped once per optimizer update. We use the AdamW optimizer with hyperparameters listed in Table 8. For learning rate $< 2e^{-4}$, we don't use linear warmup or a scheduler, we simply train the model with the constant learning rate for certain number of epochs given in Table 8.

Data Pre-processing for Arxiv: To evaluate long-context classification performance, we constructed 16K, 32K, and 64K token variants of the ArXiv dataset through a two-stage preprocessing pipeline. First, we applied a basic-English tokenizer to all documents and selected only those whose raw token count exceeded a minimum threshold, ensuring each example contained sufficiently long context. We then performed streaming sequence packing: documents were grouped by class and concatenated until reaching a target sequence length (16K, 32K, or 64K tokens), discarding only very small leftovers to maintain high packing efficiency. This produced long, contiguous sequences that preserve the natural structure of each document while enabling fixed-length training. After balancing classes, the final dataset sizes were: 16K (1947 train, 209 test), 32K (3650 train, 520 test), and 64K (3882 train, 384 test).

Table 9: Long-context image classification (ViT) performance on Food-101 (50 classes) at 16K sequence length on a 40GB A100 GPU. Train/Test denote per-epoch runtimes in seconds, and Acc. denotes accuracy. Results for Linear Attention, Linformer, and Performer were collected using batch size = 1 due to out-of-memory failures at batch size = 8. In contrast, both RACE and FlashAttention2 remain memory-efficient at this sequence length and are benchmarked with batch size = 8.

Food-101 (16K) Long-Context Image Classification			
Method	**Train** $\downarrow$	**Test** $\downarrow$	**Acc.** $\uparrow$
RACE (P=2,L=2)	**891s**	**37s**	42.4%
RACE (P=3,L=3)	950s	40s	**43.5%**
RACE (P=4,L=4)	1042s	42s	40.3%
Linear	1166s	44s	41.4%
Linformer-128	1250s	49s	20.2%
Performer-256	2546s	105s	42.4%
FlashAttention2	2600s	95s	42.1%

Table 10: **FashionMNIST @ N=784**

Method	Accuracy
RACE (P=2, L=5)	**87.7%**
RACE (P=3, L=5)	87.5%
RACE (P=4, L=4)	86.6%
RACE (P=4, L=5)	85.7%
Linformer-128	**87.7%**
FlashAttention2	87.2%
Angular (γ=8)	86.4%
Linear	85.8%
Performer-256	86.6%

Table 11: **Tiny Stories (subset) @ N=512**

Method	Perplexity
RACE (P=3, L=4)	3.9
RACE (P=4, L=4)	3.3
RACE (P=5, L=4)	**2.7**
RACE (P=5, L=5)	5.1
Linear	6.0
Angular (γ=8)	2.9
FlashAttention2	3.1
Linformer-128	4.6
Performer-256	7.1

Table 12: **Penn Tree Bank @ N=128**

Method	Test PPL
RACE (P=2, L=2)	54.7
RACE (P=3, L=3)	54.2
RACE (P=4, L=4)	**53.4**
Angular (γ=8)	58.8
Angular (γ=12)	57.6
Linear	73.2
FlashAttention2	55.4

Table 13: **Yahoo @ N=256**

Method	Accuracy
RACE (P=2, L=2)	66.9%
RACE (P=3, L=3)	66.6%
RACE (P=4, L=4)	**67.2%**
Linformer-128	64.7%
Softmax	**67.2%**
Angular (γ=8)	67.0%
Linear	66.9%
Performer-256	64.9%

Table 14: **IMDB @ N=512**

Method	Accuracy
RACE (P=2, L=2)	80.6%
RACE (P=3, L=3)	**81.3%**
RACE (P=4, L=4)	**81.3%**
Linformer-128	78.2%
FlashAttention2	80.0%
Angular (γ=6)	79.6%
Linear	80.9%
Performer-256	81.0%

Table 15: **SST-2 @ N=1024**

Method	Accuracy
RACE (P=2, L=2)	76.7%
RACE (P=4, L=4)	**79.4%**
Linformer-128	75.1%
Linear	78.0%
FlashAttention2	78.5%
Angular (γ=8)	77.2%
Performer-256	77.3%

Data Pre-processing for Food-101: To evaluate long-context image classification on Food-101 dataset with 16K context length, we discarded 50 classes as the dataset size was too large, and trained on 20K samples, and tested on 2500 samples. We use patch size = 4, image size = 512, and stride length = 4. The rest of the hyperparameters can be found in Table 8.

C Proof of Theorem 2

This section provides a complete, self-contained theoretical treatment showing that our RACE Attention closely approximates Angular Attention. We give explicit high-probability bounds for (i) the kernel error, (ii) the attention matrix error with a clean separation of numerator vs. denominator effects, and (iii) the end-to-end output error (Theorem 2). Let's also reintroduce the notations for the convenience of the reader.

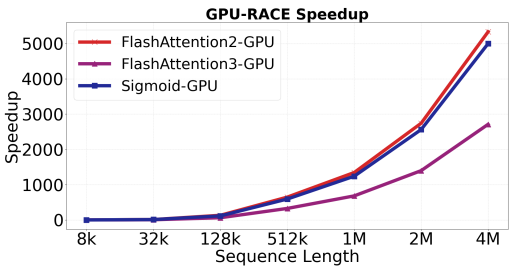


Figure 7: Speedup of RACE (ours) over self-attention kernels (FlashAttention-2/3, Sigmoid) for single-layer forward–backward on GPU; RACE is up to 2500x faster than FlashAttention-3 at 4M context length.

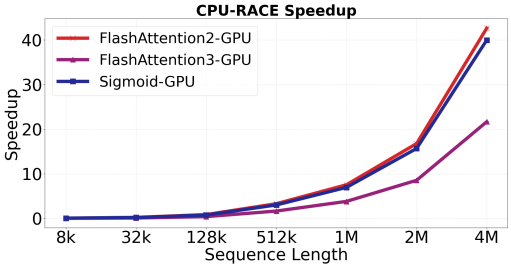


Figure 8: Speedup of RACE (ours) over self-attention kernels (FlashAttention-2/3, Sigmoid) for single-layer forward–backward on CPU; RACE is up to 20x faster than FlashAttention-3 at 4M context length.

C.1 Setup and assumptions

Data. Sequence length N, head (per-head) dimension d. Queries/keys are unit vectors:

$$Q_i, K_j \in \mathbb{R}^d \quad \text{with} \quad \|Q_i\|_2 = \|K_j\|_2 = 1, \quad i, j \in \{1, \dots, N\}.$$

Target kernel (P-powered angular).

$$\kappa(Q_i, K_j) \;:=\; \kappa_{\text{ang}}(Q_i, K_j)^P \;=\; \left(1 - \tfrac{1}{\pi}\cos^{-1}(Q_i^\top K_j)\right)^P \in [0,1], \qquad S \in \mathbb{R}^{N\times N} \text{ with } S_{ij} = \kappa(Q_i, K_j).$$

Soft RACE features. For each ensemble $\ell = 1, \dots, L$:

- Draw P random hyperplanes $W^{(\ell)} \in \mathbb{R}^{P\times d}$ whose rows $w_t^{(\ell)}$ are i.i.d.
- Corners $\mathcal{V} = \{\pm 1\}^P$ (size $R = 2^P$), with corner vectors $v_r \in \{\pm 1\}^P$.
- Logits $s^{(\ell)}(x; r) := [\tanh(W^{(\ell)} x)]^\top v_r$, temperature $\beta > 0$.
- Define the (probability) feature $\phi^{(\ell)}(x)$ by

$$[\phi^{(\ell)}(x)]_r \;=\; \frac{\exp\{\beta\, s^{(\ell)}(x; r)\}}{\sum_{r'} \exp\{\beta\, s^{(\ell)}(x; r')\}}.$$

RACE kernel and matrices. For each ensemble, define the per-table kernel matrix

$$\widehat{S}_{ij}^{(\ell)} = (\phi^{(\ell)}(Q_i))^\top (\phi^{(\ell)}(K_j)), \qquad \widehat{S} = \frac{1}{L}\sum_{\ell=1}^{L} \widehat{S}^{(\ell)}.$$

Let the (single-table) bias matrix be $\tilde{B} := \mathbb{E}[\widehat{S}^{(\ell)}] - S$.

Assumptions. For convenience, we restate the two assumptions from Sec. 3.3

(A1) Row sums of S are bounded away from zero *i.e.,* $s_{\min} := \min_i (S\mathbf{1})_i \geq C_1 N$ for some constant $C_1 > 0$, which ensures stable normalization in attention.
(A2) Spectral norm of of S is bounded *i.e.,* $\|S\|_2 \leq C_2 N$, which follows from $S_{ij} \in [0, 1]$.

Notation: We denote $\|\cdot\|_2$ as spectral norm for a matrix and Euclidean norm for a vector, $\|\cdot\|_F$ for the Frobenius norm of a matrix and for a matrix M, we denote $\|M\|_{1\to\infty} = \max_i \sum_j |M_{ij}|$.

C.2 Kernel construction with the bias term

We begin by formalizing how a single hash table induces a kernel matrix via the soft RACE features. The next lemma records norm properties that will be used repeatedly.

Lemma 3 (Bounds for a single ensemble). *Let $\Phi_Q^{(\ell)} \in \mathbb{R}^{N\times R}$ be the matrix with the i-th row $\phi^{(\ell)}(Q_i)^\top$ and $\Phi_K^{(\ell)}$ defined analogously. Then:*

1. $\widehat{S}^{(\ell)} = \Phi_Q^{(\ell)} \big(\Phi_K^{(\ell)}\big)^\top$.
2. *Each row of* $\Phi_Q^{(\ell)}$ *and* $\Phi_K^{(\ell)}$ *is a probability vector; hence* $\|\Phi_Q^{(\ell)}\|_F,\ \|\Phi_K^{(\ell)}\|_F \le \sqrt{N}$.
3. *Consequently* $\|\widehat{S}^{(\ell)}\|_F \le N$.

Proof. Each $\phi^{(\ell)}(x)$ is a softmax over $R = 2^P$ corners, so entries are nonnegative and sum to 1. Item (1) is by definition of $\widehat{S}_{ij}^{(\ell)}$. For (2), every row p satisfies $\|p\|_2 \le \|p\|_1 = 1$, hence $\|\Phi_Q^{(\ell)}\|_F^2 = \sum_i \|\phi^{(\ell)}(Q_i)\|_2^2 \le N$ (and similarly for $\Phi_K^{(\ell)}$). Item (3) follows from $\|AB\|_F \le \|A\|_F \|B\|_F$. □

Having controlled the feature-induced matrix norms, we quantify the zero-mean fluctuation of one ensemble around its expectation and prepare moment bounds needed for matrix concentration.

Lemma 4. *Let* $X^{(\ell)} := \widehat{S}^{(\ell)} - \mathbb{E}[\widehat{S}^{(\ell)}]$ *and write*

$$\Delta := \widehat{S} - S = \frac{1}{L} \sum_{\ell=1}^{L} X^{(\ell)} + \tilde{B}.$$

Then:

1. $\mathbb{E}[X^{(\ell)}] = 0$.
2. $\|X^{(\ell)}\|_2 \le 2N$.
3. *With*

$$v := \max\left\{ \left\| \sum_{\ell=1}^{L} \mathbb{E}\left[\left(\tfrac{1}{L} X^{(\ell)}\right) \left(\tfrac{1}{L} X^{(\ell)}\right)^\top \right] \right\|_2,\ \left\| \sum_{\ell=1}^{L} \mathbb{E}\left[\left(\tfrac{1}{L} X^{(\ell)}\right)^\top \left(\tfrac{1}{L} X^{(\ell)}\right) \right] \right\|_2 \right\},$$

we have $v \le 4N^2/L$.

Proof. **(1)** By definition, $X^{(\ell)} = \widehat{S}^{(\ell)} - \mathbb{E}[\widehat{S}^{(\ell)}]$, hence $\mathbb{E}[X^{(\ell)}] = \mathbb{E}[\widehat{S}^{(\ell)}] - \mathbb{E}[\widehat{S}^{(\ell)}] = 0$.

(2) By Lemma 3(3) we have $\|\widehat{S}^{(\ell)}\|_2 \le \|\widehat{S}^{(\ell)}\|_F \le N$. By convexity of the spectral norm,

$$\|\mathbb{E}[\widehat{S}^{(\ell)}]\|_2 \le \mathbb{E}\big[\|\widehat{S}^{(\ell)}\|_2\big] \le N.$$

Therefore, by the triangle inequality,

$$\|X^{(\ell)}\|_2 \ = \ \|\widehat{S}^{(\ell)} - \mathbb{E}[\widehat{S}^{(\ell)}]\|_2 \ \le \ \|\widehat{S}^{(\ell)}\|_2 + \|\mathbb{E}[\widehat{S}^{(\ell)}]\|_2 \ \le \ 2N.$$

(3) Let $Y^{(\ell)} := \frac{1}{L} X^{(\ell)}$. Then

$$\sum_{\ell=1}^{L} \mathbb{E}\big[Y^{(\ell)} (Y^{(\ell)})^\top\big] = \frac{1}{L^2} \sum_{\ell=1}^{L} \mathbb{E}\big[X^{(\ell)} (X^{(\ell)})^\top\big].$$

Using subadditivity of $\|\cdot\|_2$, Jensen, and $\|AB\|_2 \le \|A\|_2 \|B\|_2$,

$$\left\| \sum_{\ell=1}^{L} \mathbb{E}\big[Y^{(\ell)} (Y^{(\ell)})^\top\big] \right\|_2 \le \frac{1}{L^2} \sum_{\ell=1}^{L} \left\| \mathbb{E}\big[X^{(\ell)} (X^{(\ell)})^\top\big] \right\|_2 \le \frac{1}{L^2} \sum_{\ell=1}^{L} \mathbb{E}\big[\|X^{(\ell)}\|_2^2\big] \le \frac{1}{L^2} \sum_{\ell=1}^{L} (2N)^2 = \frac{4N^2}{L}.$$

The same bound holds for $\left\| \sum_{\ell=1}^{L} \mathbb{E}\big[(Y^{(\ell)})^\top Y^{(\ell)}\big] \right\|_2$ by symmetry. Taking the maximum of the two yields $v \le 4N^2/L$. □

To convert the moment and uniform bounds above into a high-probability spectral-norm bound, we invoke a standard matrix Bernstein inequality from Tropp (2015) , stated next for completeness.

Lemma 5 (Matrix Bernstein). *If $Z^{(\ell)} \in \mathbb{R}^{m\times n}$ are independent mean-zero matrices with $\|Z^{(\ell)}\|_2 \le H$ and variance proxy v, then for any $t > 0$,*

$$\mathbb{P}\left(\left\|\sum_{\ell} Z^{(\ell)}\right\|_2 \ge t\right) \le (m+n)\,\exp\left(-\frac{t^2/2}{v + Ht/3}\right).$$

Next, applying Lemma 5 with the parameters established in Lemma 4, we obtain the following nonasymptotic deviation bound for the kernel estimator.

Theorem 6 (Kernel deviation with explicit constants). *With probability at least $1-\delta$,*

$$\|\widehat{S} - S\|_2 \;\le\; \|\tilde{B}\|_2 \;+\; 4\,\frac{N}{\sqrt{L}}\sqrt{\log\frac{2N}{\delta}} \;+\; \frac{4}{3}\,\frac{N}{L}\,\log\frac{2N}{\delta}.$$

Proof. First, rewrite $\widehat{S} - S$ as

$$\widehat{S} - S = \frac{1}{L}\sum_{\ell=1}^{L}\widehat{S}^{(\ell)} - S = \frac{1}{L}\sum_{\ell=1}^{L}\left(\widehat{S}^{(\ell)} - \mathbb{E}[\widehat{S}^{(\ell)}]\right) \;+\; \left(\mathbb{E}[\widehat{S}^{(\ell)}] - S\right) = \frac{1}{L}\sum_{\ell=1}^{L} X^{(\ell)} + \tilde{B}.$$

By the triangle inequality,

$$\|\widehat{S} - S\|_2 \;\le\; \left\|\frac{1}{L}\sum_{\ell=1}^{L} X^{(\ell)}\right\|_2 \;+\; \|\tilde{B}\|_2.$$

It remains to upper bound the random term with high probability.

Set $Z^{(\ell)} := \frac{1}{L}X^{(\ell)}$. Then the $Z^{(\ell)}$ are independent, mean-zero, $N \times N$ random matrices. From Lemma 4(2) we have $\|X^{(\ell)}\|_2 \le 2N$. Therefore, $\|Z^{(\ell)}\|_2 \le H := \frac{2N}{L}$. Similarly, Lemma 4(3) gives $v \le \frac{4N^2}{L}$. Applying Lemma 5 with $m = n = N$ yields

$$\mathbb{P}\left(\left\|\sum_{\ell=1}^{L} Z^{(\ell)}\right\|_2 \ge t\right) \;\le\; 2N\,\exp\left(-\frac{t^2}{2\,(v + Ht/3)}\right).$$

Let $u := \log\frac{2N}{\delta}$. To make the RHS $\le \delta$, it suffices that

$$\frac{t^2}{2\,(v + Ht/3)} \;\ge\; u \quad\Longleftrightarrow\quad t^2 - \frac{2uH}{3}\,t - 2u\,v \;\ge\; 0.$$

Choose

$$t \;=\; 2\sqrt{v\,u} \;+\; \frac{2}{3}\,H\,u.$$

Writing $a := 2\sqrt{vu}$ and $b := \frac{2}{3}Hu$ (so $t = a + b$) gives

$$t^2 - \frac{2uH}{3}\,t - 2uv = (a+b)^2 - \frac{2uH}{3}(a+b) - 2uv = (4vu - 2uv) + \left(\tfrac{8}{3} - \tfrac{4}{3}\right)Hu\sqrt{vu} \ge 0.$$

Therefore,

$$\left\|\sum_{\ell=1}^{L} Z^{(\ell)}\right\|_2 \;\le\; 2\sqrt{v\,u} \;+\; \frac{2}{3}\,H\,u \quad \text{with probability at least } 1-\delta.$$

Plugging $v \le \frac{4N^2}{L}$ and $H = \frac{2N}{L}$ yields

$$\left\|\sum_{\ell=1}^{L} Z^{(\ell)}\right\|_2 \;\le\; 2\sqrt{\frac{4N^2}{L}\,u} \;+\; \frac{2}{3}\cdot\frac{2N}{L}\,u = 4\,\frac{N}{\sqrt{L}}\,\sqrt{u} \;+\; \frac{4}{3}\,\frac{N}{L}\,u.$$

Since $\sum_{\ell=1}^{L} Z^{(\ell)} = \frac{1}{L}\sum_{\ell=1}^{L} X^{(\ell)}$, we conclude that

$$\left\|\frac{1}{L}\sum_{\ell=1}^{L} X^{(\ell)}\right\|_2 \;\le\; 4\,\frac{N}{\sqrt{L}}\,\sqrt{\log\frac{2N}{\delta}} \;+\; \frac{4}{3}\,\frac{N}{L}\,\log\frac{2N}{\delta} \quad \text{with probability } \ge 1-\delta,$$

and therefore

$$\|\widehat{S} - S\|_2 \;\le\; \|\tilde{B}\|_2 \;+\; 4\,\frac{N}{\sqrt{L}}\sqrt{\log\frac{2N}{\delta}} \;+\; \frac{4}{3}\,\frac{N}{L}\,\log\frac{2N}{\delta},$$

as claimed. □

The deviation bound decomposes into a variance term and a (deterministic) bias term $\tilde{B}$. We now bound $\tilde{B}$ explicitly as a function of β and P.

Lemma 7 (Bias to the P-powered angular kernel: explicit bound)**.** *Fix $P \ge 1$ and $\beta > 0$. With S and $\tilde{B}$ as above, let $c := 2\tanh(1)$ and $C_1 := \frac{2}{\sqrt{2\pi}}e^{-1/2}$. Then*

$$\|\tilde{B}\|_2 \;\le\; \frac{4}{\sqrt{2\pi}}\,\frac{NP}{\beta} \;+\; \underbrace{\left(\tfrac{4}{\sqrt{2\pi}}e^{-1/2}\right)}_{=2C_1} NP\,e^{-c\beta}.$$

In particular, as $\beta \to \infty$,

$$\|\tilde{B}\|_2 \;=\; \frac{4}{\sqrt{2\pi}}\,\frac{NP}{\beta} \;+\; o\Big(\tfrac{N}{\beta}\Big).$$

Proof. Let us denote the inner product similarity by $\rho := Q_i^\top K_j$, and recall that the angular kernel is $\kappa_{\mathrm{ang}}(Q_i, K_j) := 1 - \frac{1}{\pi}\cos^{-1}(\rho)$ From standard LSH theory, for P i.i.d. Gaussian hyperplanes $W^{(\ell)} \in \mathbb{R}^{P\times d}$, the probability that all P bits match is exactly:

$$\mathbb{P}\left(h_P(Q_i) = h_P(K_j)\right) = \kappa_{\mathrm{ang}}(Q_i, K_j)^P = S_{ij}.$$

Now define the softmax sketch feature for the hash table ℓ:

$$s^{(\ell)}(x; r) := \tanh(W^{(\ell)}x)^\top v_r, \qquad [\phi^{(\ell)}(x)]_r := \frac{e^{\beta s^{(\ell)}(x;r)}}{\sum_{r'} e^{\beta s^{(\ell)}(x;r')}},$$

where $v_r \in \{\pm 1\}^P$ denotes the binary corner vectors of length P, and $R = 2^P$.

Let $\widehat{S}^{(\ell)} \in \mathbb{R}^{N\times N}$ be the kernel matrix for a single hash table:

$$\widehat{S}^{(\ell)}_{ij} := \mathbb{E}\left[\left(\phi^{(\ell)}(Q_i)\right)^\top\left(\phi^{(\ell)}(K_j)\right)\right], \qquad S_{ij} := \kappa_{\mathrm{ang}}(Q_i, K_j)^P,$$

and recall the bias matrix is $\tilde{B} := \mathbb{E}[\widehat{S}^{(\ell)}] - S$.

Our goal is to bound $\|\tilde{B}\|_2$. To do this, fix any pair (i, j) and note:

$$|\widehat{S}^{(\ell)}_{ij} - S_{ij}| = \left|\mathbb{E}\left[\left(\phi^{(\ell)}(Q_i)\right)^\top\left(\phi^{(\ell)}(K_j)\right)\right] - \mathbb{P}[h_P(Q_i) = h_P(K_j)]\right|.$$

Let $r^\star(Q_i) := \arg\max_r s^{(\ell)}(Q_i; r)$ be the corner with highest logit for Q_i, and similarly for K_j. Then, from the standard softmax tail bound:

$$1 - [\phi^{(\ell)}(x)]_{r^\star(x)} = \sum_{r\neq r^\star(x)} [\phi^{(\ell)}(x)]_r = \sum_{r\neq r^\star(x)} \frac{e^{\beta s^{(\ell)}(x;r)}}{\sum_{r'} e^{\beta s^{(\ell)}(x;r')}}$$

$$\le \sum_{r\neq r^\star(x)} \frac{e^{\beta s^{(\ell)}(x;r)}}{e^{\beta s^{(\ell)}(x;r^\star(x))}} = \sum_{r\neq r^\star(x)} \exp\Big\{-\beta\big(s^{(\ell)}(x; r^\star(x)) - s^{(\ell)}(x; r)\big)\Big\},$$

where the inequality follows from replacing the sum in the denominator just by one term. Since $s^{(\ell)}(x; r) = \sum_{t=1}^P u_t(x)\cdot r_t$, where $u_t(x) := \tanh(w_t^\top x)$ and w_t is the t-th row of $W^{(\ell)}$, any single-bit flip from $r^\star$ to r changes the score by at least $2|u_t(x)|$. There are P such flips, so:

$$1 - [\phi^{(\ell)}(x)]_{r^\star(x)} \le \sum_{t=1}^P \exp(-2\beta|u_t(x)|). \tag{5}$$

Now, for each bit, $u_t(x) = \tanh(w_t^\top x)$ with $w_t^\top x \sim \mathcal{N}(0,1)$. Let $Z \sim \mathcal{N}(0,1)$. Then

$$\mathbb{E}[e^{-2\beta|u_t(x)|}] = \mathbb{E}[e^{-2\beta|\tanh(Z)|}].$$

We split into two regions. (1) On $|Z| \le 1$, we use $|\tanh z| \ge \frac{|z|}{2}$, so $e^{-2\beta|\tanh(Z)|} \le e^{-\beta|Z|}$. Hence

$$\mathbb{E}[e^{-2\beta|\tanh(Z)|}\mathbf{1}_{|Z|\le 1}] \le \frac{2}{\sqrt{2\pi}} \int_0^1 e^{-\beta z} e^{-z^2/2}\, dz \le \frac{2}{\sqrt{2\pi}} \cdot \frac{1}{\beta}.$$

(2) On $|Z| > 1$, we use $\tanh z \ge \tanh(1)$, so $e^{-2\beta|\tanh(Z)|} \le e^{-2\beta\tanh(1)}$. Thus

$$\mathbb{E}[e^{-2\beta|\tanh(Z)|}\mathbf{1}_{|Z|>1}] \le e^{-2\beta\tanh(1)}\mathbb{P}(|Z|>1) = 2e^{-2\beta\tanh(1)}\,\mathbb{P}(Z>1) \le e^{-2\beta\tanh(1)}\sqrt{\frac{2}{\pi}}e^{-1/2} \le e^{-c\beta},$$

where the second last bound is due to Mill's inequality and the last inequality follows from the fact that $\sqrt{\frac{2}{\pi}}e^{-1/2} \le 1$. Here $c = 2\tanh(1)$.

Combining the two expectations we get,

$$\mathbb{E}[e^{-2\beta|u_t(x)|}] \le \frac{2}{\sqrt{2\pi}\,\beta} + e^{-c\beta}, \quad c = 2\tanh(1).$$

Substituting back to eqn. 5,

$$\mathbb{E}[1 - [\phi^{(\ell)}(x)]_{r^\star(x)}] \;\le\; \frac{2P}{\sqrt{2\pi}\,\beta} + O(Pe^{-c\beta}).$$

Applying this to both Q_i and K_j gives

$$|\widehat{S}_{ij}^{(\ell)} - S_{ij}| \;\le\; \frac{4P}{\sqrt{2\pi}\,\beta} + O(Pe^{-c\beta}).$$

Therefore,

$$\|\tilde{B}\|_2 \le \|\tilde{B}\|_F \le N \sup_{i,j} |\tilde{B}_{ij}| \;\le\; N\left(\frac{4P}{\sqrt{2\pi}\,\beta} + O(Pe^{-c\beta})\right).$$

Equivalently,

$$\|\tilde{B}\|_2 \;\le\; \frac{4}{\sqrt{2\pi}}\,\frac{NP}{\beta} + o\Big(\frac{N}{\beta}\Big),$$

since the exponential term is negligible compared to $1/\beta$. This proves the claim. □

We now propagate the kernel-level error into the attention matrix. This requires controlling the normalization (row sums) and its inverse, which we address next.

C.3 From kernels to attention (numerator vs. denominator)

Write $\widehat{S} = S + \Delta$, $D = \mathrm{diag}(S\mathbf{1})$, and $\widehat{D} = \mathrm{diag}(\widehat{S}\mathbf{1}) = D + E$. Define the attention matrices

$$A := D^{-1}S, \qquad \widehat{A} := \widehat{D}^{-1}\widehat{S}.$$

The following lemma ties the row-sum perturbation E to Δ and gives a simple invertibility condition for $\widehat{D}$.

Lemma 8 (Row-sum and inverse diagonal control)**.** *Recall that $s_{\min} = \min_i D_{ii} > 0$. Then*

1. $\|E\|_2 \le \|\Delta\|_{1\to\infty} \le \sqrt{N}\,\|\Delta\|_2$.
2. *If $\|E\|_2 \le s_{\min}/2$, then $\|\widehat{D}^{-1}\|_2 \le 2/s_{\min}$.*

Proof. **(1) Row-sum control.** Since $\widehat{S} = S + \Delta$ and $\widehat{D} = \mathrm{diag}(\widehat{S}\mathbf{1})$,

$$\text{Let } E \;:=\; \widehat{D} - D \;=\; \mathrm{diag}\big((\widehat{S} - S)\mathbf{1}\big) \;=\; \mathrm{diag}(\Delta\mathbf{1}).$$

Hence each diagonal entry is $E_{ii} = (\Delta\mathbf{1})_i = \sum_{j=1}^{N} \Delta_{ij}$, so

$$\|E\|_2 \;=\; \max_i |E_{ii}| \;=\; \max_i \big|(\Delta\mathbf{1})_i\big| \;\le\; \max_i \sum_{j=1}^{N} |\Delta_{ij}| \;=\; \|\Delta\|_{1\to\infty}.$$

For the second inequality, by Cauchy–Schwarz on each row i,

$$\sum_{j=1}^{N} |\Delta_{ij}| \;\le\; \sqrt{N}\Big(\sum_{j=1}^{N} \Delta_{ij}^2\Big)^{1/2} \;=\; \sqrt{N}\,\|\Delta_{i,\cdot}\|_2.$$

Taking the maximum over i and using $\max_i \|\Delta_{i,\cdot}\|_2 \le \|\Delta\|_F \le \sqrt{N}\,\|\Delta\|_2$ gives

$$\|\Delta\|_{1\to\infty} \;=\; \max_i \sum_j |\Delta_{ij}| \;\le\; \sqrt{N}\,\max_i \|\Delta_{i,\cdot}\|_2 \;\le\; \sqrt{N}\,\|\Delta\|_2.$$

(2) Inverse diagonal control. Because $\widehat{D} = D + E$ is diagonal, its smallest diagonal entry satisfies

$$\min_i \widehat{D}_{ii} \;=\; \min_i (D_{ii} + E_{ii}) \;\ge\; \min_i D_{ii} - \max_i |E_{ii}| \;=\; s_{\min} - \|E\|_2.$$

If $\|E\|_2 \le s_{\min}/2$, then $\min_i \widehat{D}_{ii} \ge s_{\min}/2 > 0$, so $\widehat{D}$ is invertible and

$$\|\widehat{D}^{-1}\|_2 \;=\; \max_i \frac{1}{\widehat{D}_{ii}} \;\le\; \frac{1}{s_{\min} - \|E\|_2} \;\le\; \frac{1}{s_{\min} - s_{\min}/2} \;=\; \frac{2}{s_{\min}}.$$

□

Assumption (A1) ensures D has diagonals of order N, but we must still control E. The next lemma shows that the condition $\|E\|_2 \le s_{\min}/2$ holds with high probability once L is moderately large.

Lemma 9 (Concentration bound for E)**.** *Under assumption (A1), with probability at least* $1 - \delta$*,*

$$\|E\|_2 \;\le\; \tfrac{1}{2}\, s_{\min}$$

provided that

$$L \;\ge\; \frac{2}{C_1^2}\,\log\frac{2N^2}{\delta}.$$

Proof. Recall $E = \widehat{D} - D = \mathrm{diag}(\Delta\mathbf{1})$ with $\Delta = \widehat{S} - S$. Hence

$$\|E\|_2 \;=\; \max_i |(\Delta\mathbf{1})_i| = \max_i \Big|\sum_{j=1}^{N} (\widehat{S}_{ij} - S_{ij})\Big|.$$

Each entry $\widehat{S}_{ij}$ is the average of L i.i.d. bounded random variables $\widehat{S}_{ij}^{(\ell)} \in [0,1]$ with mean S_{ij}. By Hoeffding's inequality,

$$\Pr\big(|\widehat{S}_{ij} - S_{ij}| > \epsilon\big) \;\le\; 2\exp(-2L\epsilon^2).$$

A union bound over all N^2 pairs (i,j) gives

$$\Pr\Big(\max_{i,j} |\widehat{S}_{ij} - S_{ij}| > \epsilon\Big) \;\le\; 2N^2\exp(-2L\epsilon^2).$$

Thus with probability at least $1 - \delta$,

$$\max_{i,j} |\widehat{S}_{ij} - S_{ij}| \;\le\; \sqrt{\tfrac{1}{2L}\,\log\tfrac{2N^2}{\delta}}.$$

For any row i,

$$\Big|\sum_{j=1}^{N}(\widehat{S}_{ij}-S_{ij})\Big| \;\le\; N\,\max_j|\widehat{S}_{ij}-S_{ij}|,$$

so with probability $\ge 1-\delta$,

$$\|E\|_2 \;\le\; N\sqrt{\tfrac{1}{2L}\,\log\tfrac{2N^2}{\delta}}.$$

By (A1), $s_{\min}\ge C_1N$. Therefore $\|E\|_2\le\frac{1}{2}s_{\min}$ whenever

$$N\sqrt{\tfrac{1}{2L}\,\log\tfrac{2N^2}{\delta}} \;\le\; \tfrac{1}{2}C_1N,$$

which simplifies to the claimed condition $L\ge\frac{2}{C_1^2}\,\log\frac{2N^2}{\delta}$. □

Now, with row-sums controlled, we relate $\widehat{A}$ and A exactly through a decomposition that isolates the contributions of Δ in both the numerator and denominator.

Lemma 10 (Exact perturbation identity and bound)**.**

$$\widehat{A}-A=D^{-1}\Delta+\big(\widehat{D}^{-1}-D^{-1}\big)S+\widehat{D}^{-1}\Delta.$$

Moreover, whenever $\|E\|_2<s_{\min}$,

$$\|\widehat{A}-A\|_2\le\frac{\|\Delta\|_2}{s_{\min}}+\frac{\|S\|_2\|E\|_2}{s_{\min}(s_{\min}-\|E\|_2)}+\frac{\|\Delta\|_2}{s_{\min}-\|E\|_2}.$$

Proof. Recall $\widehat{S}=S+\Delta$, $\widehat{D}=D+E$, $A=D^{-1}S$, $\widehat{A}=\widehat{D}^{-1}\widehat{S}$. We obtain the identity by adding and subtracting $D^{-1}\widehat{S}$ and $\widehat{D}^{-1}S$:

$$\begin{aligned}\widehat{A}-A&=\widehat{D}^{-1}\widehat{S}-D^{-1}S\\&=\underbrace{\big(\widehat{D}^{-1}\widehat{S}-\widehat{D}^{-1}S\big)}_{=\,\widehat{D}^{-1}(\widehat{S}-S)=\widehat{D}^{-1}\Delta}+\underbrace{\big(\widehat{D}^{-1}S-D^{-1}S\big)}_{=(\widehat{D}^{-1}-D^{-1})S}\\&=\widehat{D}^{-1}\Delta+(\widehat{D}^{-1}-D^{-1})S.\end{aligned}$$

Next, write

$$(\widehat{D}^{-1}-D^{-1})S=\widehat{D}^{-1}\big(I-\widehat{D}D^{-1}\big)S\;=\;-\widehat{D}^{-1}\big(\widehat{D}-D\big)D^{-1}S\;=\;-\widehat{D}^{-1}E\,D^{-1}S.$$

Adding and subtracting $D^{-1}\Delta$ yields the displayed identity:

$$\widehat{A}-A=D^{-1}\Delta\;+\;(\widehat{D}^{-1}-D^{-1})S\;+\;\widehat{D}^{-1}\Delta.$$

For the bound, use submultiplicativity and Lemma 8. When $\|E\|_2<s_{\min}$, we have $\|D^{-1}\|_2=1/s_{\min}$ and $\|\widehat{D}^{-1}\|_2\le1/(s_{\min}-\|E\|_2)$. Moreover,

$$\|\widehat{D}^{-1}-D^{-1}\|_2=\|\widehat{D}^{-1}(\,D-\widehat{D}\,)D^{-1}\|_2\le\|\widehat{D}^{-1}\|_2\,\|E\|_2\,\|D^{-1}\|_2\le\frac{\|E\|_2}{s_{\min}\big(s_{\min}-\|E\|_2\big)}.$$

Therefore,

$$\begin{aligned}\|\widehat{A}-A\|_2&\le\|D^{-1}\|_2\,\|\Delta\|_2\;+\;\|\widehat{D}^{-1}-D^{-1}\|_2\,\|S\|_2\;+\;\|\widehat{D}^{-1}\|_2\,\|\Delta\|_2\\&\le\frac{\|\Delta\|_2}{s_{\min}}\;+\;\frac{\|S\|_2\,\|E\|_2}{s_{\min}\big(s_{\min}-\|E\|_2\big)}\;+\;\frac{\|\Delta\|_2}{s_{\min}-\|E\|_2},\end{aligned}$$

which proves the claim. □

Specializing Lemma 10 to the regime $\|E\|_2\le s_{\min}/2$ (guaranteed w.h.p. by Lemma 9), we obtain a concise spectral bound for $\|\widehat{A}-A\|_2$ in terms of $\|\Delta\|_2$.

Lemma 11 (Attention deviation). *If $\|E\|_2 \le s_{\min}/2$, then*

$$\|\widehat{A} - A\|_2 \le \frac{2\|\Delta\|_2}{s_{\min}} + \frac{2\|S\|_2}{s_{\min}^2}\sqrt{N}\,\|\Delta\|_2.$$

Proof. From Lemma 10,

$$\|\widehat{A} - A\|_2 \;\le\; \frac{\|\Delta\|_2}{s_{\min}} + \frac{\|S\|_2\|E\|_2}{s_{\min}(s_{\min} - \|E\|_2)} + \frac{\|\Delta\|_2}{s_{\min} - \|E\|_2}.$$

Since $\|E\|_2 \le s_{\min}/2$, it follows that

$$\frac{1}{s_{\min} - \|E\|_2} \;\le\; \frac{1}{s_{\min}/2} = \frac{2}{s_{\min}}.$$

Substituting this bound gives

$$\|\widehat{A} - A\|_2 \;\le\; \frac{\|\Delta\|_2}{s_{\min}} + \frac{2\|S\|_2}{s_{\min}^2}\,\|E\|_2 + \frac{2\|\Delta\|_2}{s_{\min}}.$$

By Lemma 8(1), $\|E\|_2 \le \|\Delta\|_{1\to\infty} \le \sqrt{N}\,\|\Delta\|_2$. Therefore,

$$\|\widehat{A} - A\|_2 \;\le\; \frac{2\|\Delta\|_2}{s_{\min}} + \frac{2\|S\|_2}{s_{\min}^2}\sqrt{N}\,\|\Delta\|_2,$$

which proves the claim. □

Finally, we translate attention deviation into end-to-end output deviation by a single multiplication with the value matrix V, yielding the main finite-sample guarantee.

Theorem 12 (End-to-end output error). *Let $V \in \mathbb{R}^{N\times d}$ be the value matrix. With probability at least $1-\delta$, if $\|E\|_2 \le s_{\min}/2$ then*

$$\|\widehat{O} - O\|_F \;\le\; \left(\tfrac{2}{s_{\min}} + \tfrac{2\|S\|_2\sqrt{N}}{s_{\min}^2}\right)\left(\frac{4}{\sqrt{2\pi}}\,\frac{NP}{\beta} + O\big(NPe^{-c\beta}\big) + 4\,\frac{N}{\sqrt{L}}\sqrt{\log\tfrac{2N}{\delta}} + \tfrac{4}{3}\,\frac{N}{L}\,\log\tfrac{2N}{\delta}\right)\|V\|_F,$$

where $c = 2\tanh(1)$.

Proof. By the estimator identity, $\widehat{O} = \widehat{A}V$ and $O = AV$, hence

$$\|\widehat{O} - O\|_F = \|(\widehat{A} - A)V\|_F \le \|\widehat{A} - A\|_2\,\|V\|_F.$$

Under the condition $\|E\|_2 \le s_{\min}/2$, Lemma 11 (Attention deviation) gives

$$\|\widehat{A} - A\|_2 \;\le\; \left(\tfrac{2}{s_{\min}} + \tfrac{2\|S\|_2\sqrt{N}}{s_{\min}^2}\right)\|\widehat{S} - S\|_2.$$

Applying Theorem 6 (Kernel deviation) yields, with probability at least $1-\delta$,

$$\|\widehat{S} - S\|_2 \;\le\; \|\tilde{B}\|_2 \;+\; 4\,\frac{N}{\sqrt{L}}\sqrt{\log\frac{2N}{\delta}} \;+\; \frac{4}{3}\,\frac{N}{L}\,\log\frac{2N}{\delta}.$$

Finally, substitute the explicit bias bound from Lemma 7:

$$\|\tilde{B}\|_2 \;\le\; \frac{4}{\sqrt{2\pi}}\,\frac{NP}{\beta} \;+\; O\big(NPe^{-c\beta}\big), \qquad c = 2\tanh(1).$$

Combining the three displays proves the stated inequality:

$$\|\widehat{O} - O\|_F \;\le\; \left(\tfrac{2}{s_{\min}} + \tfrac{2\|S\|_2\sqrt{N}}{s_{\min}^2}\right)\left(\frac{4}{\sqrt{2\pi}}\,\frac{NP}{\beta} + O\big(NPe^{-c\beta}\big) + 4\,\frac{N}{\sqrt{L}}\sqrt{\log\tfrac{2N}{\delta}} + \tfrac{4}{3}\,\frac{N}{L}\,\log\tfrac{2N}{\delta}\right)\|V\|_F.$$

□

Proof of Theorem 2. From Theorem 12 we have, with probability at least $1-\delta$ and on the event $\|E\|_2 \le s_{\min}/2$,

$$\|\widehat{O} - O\|_F \;\le\; \left(\tfrac{2}{s_{\min}} + \tfrac{2\|S\|_2\sqrt{N}}{s_{\min}^2}\right)\left(\|\tilde{B}\|_2 + 4\,\frac{N}{\sqrt{L}}\sqrt{\log\tfrac{2N}{\delta}} + \tfrac{4}{3}\,\frac{N}{L}\,\log\tfrac{2N}{\delta}\right)\|V\|_F.$$

By the explicit bias bound (Lemma 7), $\|\tilde{B}\|_2 \le \frac{4}{\sqrt{2\pi}}\frac{NP}{\beta} + O(NPe^{-c\beta})$ with $c = 2\tanh(1)$. Under (A1) and (A2), $s_{\min} \ge C_1 N$ and $\|S\|_2 \le C_2 N$, so

$$\frac{2}{s_{\min}} + \frac{2\|S\|_2\sqrt{N}}{s_{\min}^2} \;\le\; \frac{2}{C_1 N} + \frac{2C_2 N\sqrt{N}}{C_1^2 N^2} \;=\; O\Big(\frac{1}{\sqrt{N}}\Big).$$

Plugging these into the bound gives

$$\|\widehat{O} - O\|_F \;\le\; O\Big(\tfrac{1}{\sqrt{N}}\Big)\left(\tfrac{NP}{\beta} + \tfrac{N}{\sqrt{L}}\sqrt{\log\tfrac{2N}{\delta}} + \tfrac{N}{L}\log\tfrac{2N}{\delta} + NPe^{-c\beta}\right)\|V\|_F.$$

Hence

$$\|\widehat{O} - O\|_F \;\le\; O\Big(\tfrac{P}{\beta} + \sqrt{\tfrac{\log(2N/\delta)}{L}} + \tfrac{1}{\sqrt{N}}\,\tfrac{N}{L}\log\tfrac{2N}{\delta} + \tfrac{P}{\beta}\sqrt{N}\,e^{-c\beta}\Big)\|V\|_F.$$

The exponentially small term is negligible for moderate β. Moreover, for $L \gtrsim \log N$ the middle $\frac{N}{L}$ term is dominated by the $\sqrt{\frac{\log(2N/\delta)}{L}}$ term. Absorbing absolute constants and the mild $\log(2N/\delta)$ vs. $\log(N/\delta)$ difference into the Big-$\mathcal{O}$, we obtain

$$\|\widehat{O} - O\|_F \;=\; \mathcal{O}\left(\left(\tfrac{P}{\beta} + \sqrt{\tfrac{\log(N/\delta)}{L}}\right)\|V\|_F\right),$$

with probability at least $1-\delta$, which proves the theorem. □

D CAUSAL RACE ATTENTION

Algorithm 2 RACE Attention (causal)

Input: $Q, K, V \in \mathbb{R}^{N\times d}$; number of hash tables L; number of hyperplanes P; temperature $\beta > 0$.
Output: $\widehat{O} \in \mathbb{R}^{N\times d}$.

1: **for** $\ell = 1, \dots, L$ **do**
2: Draw $W^{(\ell)} \in \mathbb{R}^{P\times d}$ // P random hyperplanes
3: Define the corner set $\mathcal{V} = \{\pm 1\}^P$ ($R = 2^P$) with $v_r \in \{\pm 1\}^P$ // R corners
4: Build $\Phi_Q^{(\ell)}, \Phi_K^{(\ell)} \in \mathbb{R}^{N\times R}$ with rows

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

5: Initialize cumulative bucket statistics:

$$A_{\text{cum}}^{(\ell)} \leftarrow \mathbf{0}_R \in \mathbb{R}^R, \qquad B_{\text{cum}}^{(\ell)} \leftarrow \mathbf{0}_{R\times d} \in \mathbb{R}^{R\times d}.$$

6: **for** $t = 1, \dots, N$ **do**
7: $\Phi_K^{(\ell)}[t,:] \in \mathbb{R}^R, \quad V_t \in \mathbb{R}^d$
8: $A_{\text{cum}}^{(\ell)} \leftarrow A_{\text{cum}}^{(\ell)} + (\Phi_K^{(\ell)}[t,:])^\top$ // $\mathbb{R}^R$
9: $B_{\text{cum}}^{(\ell)} \leftarrow B_{\text{cum}}^{(\ell)} + (\Phi_K^{(\ell)}[t,:])^\top V_t$ // $\mathbb{R}^{R\times d}$
10: $\Phi_Q^{(\ell)}[t,:] \in \mathbb{R}^R$
11: $\texttt{num}_t^{(\ell)} \leftarrow \Phi_Q^{(\ell)}[t,:]\; B_{\text{cum}}^{(\ell)}$ // $(1\times R)\cdot(R\times d) = \mathbb{R}^d$
12: $\texttt{den}_t^{(\ell)} \leftarrow \Phi_Q^{(\ell)}[t,:]\; A_{\text{cum}}^{(\ell)}$ // $(1\times R)\cdot(R) = \mathbb{R}$
13: **end for**
14: **end for**
15: For each $t = 1, \dots, N$:

$$\texttt{Num}_t = \frac{1}{L}\sum_{\ell=1}^{L}\texttt{num}_t^{(\ell)} \in \mathbb{R}^d, \qquad \texttt{Den}_t = \frac{1}{L}\sum_{\ell=1}^{L}\texttt{den}_t^{(\ell)} \in \mathbb{R}, \qquad \widehat{O}_t = \frac{\texttt{Num}_t}{\texttt{Den}_t} \in \mathbb{R}^d.$$

16: **return** $\widehat{O} = \begin{bmatrix}\widehat{O}_1^\top \\ \vdots \\ \widehat{O}_N^\top\end{bmatrix} \in \mathbb{R}^{N\times d}$.

We implemented the causal version (see Algorithm 2) efficiently using OpenMP-based parallelization rather than a naive nested-loop approach. Each hash table is processed in a separate thread with its own cumulative bucket arrays, and updates are performed incrementally in a single left-to-right scan. This avoids redundant recomputation at every step using torch.cumsum() and enables CPU-level parallel execution with negligible synchronization overhead.

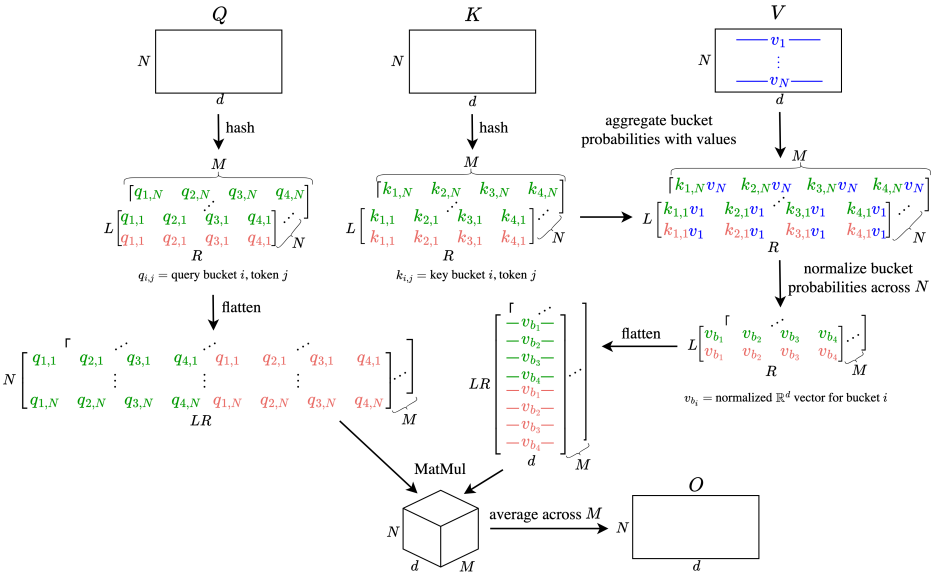


Figure 9: RACE Attention pipeline from the inputs $Q, K, V \in R^{N \times d}$ to the output $O \in R^{N \times d}$: queries/keys are soft-hashed into R buckets across L tables and M ensembles, keys/values form per-bucket summaries, and each query mixes the matched summaries to produce O.

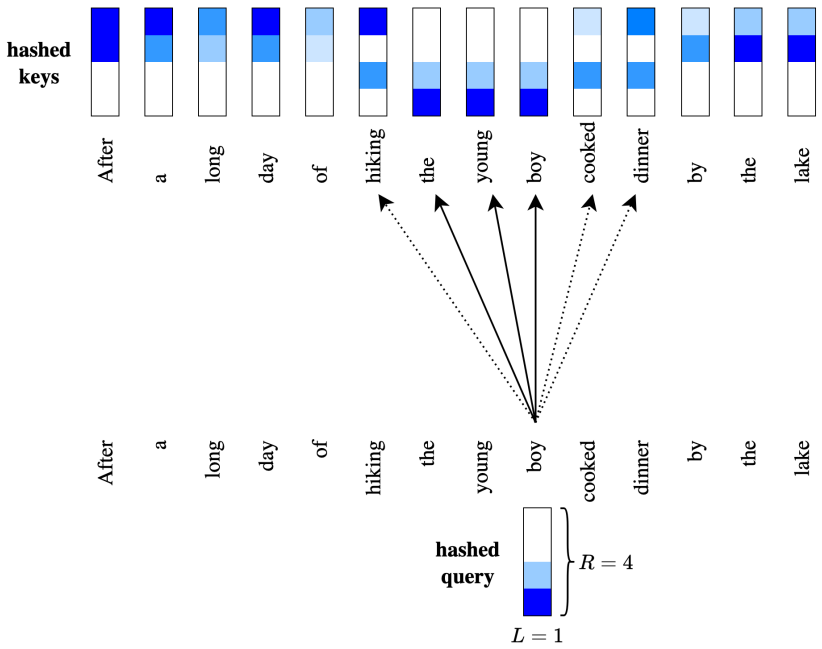


Figure 10: An intuitive schematic of how RACE Attention runs with L hash tables and R buckets per table. Similarity between Queries and Keys is highest if they both hash to same buckets across all hash tables.