

Lectures 6 & 7: Algorithmic Concepts in Attention Mechanisms

Instructors: *Alex Andoni, Edo Liberty*Lecturer: *Edo Liberty*

Lecture Overview & Pedagogical Roadmap

As Large Language Models (LLMs) scale to context windows of $n = 128\text{k}$ to 1M tokens, **scaled dot-product attention** becomes the primary computational and memory bottleneck. While hardware-aware kernels (FlashAttention) optimize GPU memory traffic, they still execute all $\Theta(n^2d)$ arithmetic operations.

These notes develop the core algorithmic tools for sub-quadratic attention across two 90-minute sessions:

1. **Session 1 — Attention Bottlenecks, The Kernel Trick, and Linearization (Sections 1–3):** After quantifying the exact FLOP and KV-cache walls in modern LLMs and surveying ANN/MIPS approaches to sparse attention, we study the **Kernel Trick** (motivated via Kernel SVMs) and **Random Fourier Features** [8]. Approximating the softmax kernel by explicit low-dimensional features $\phi : \mathbb{R}^d \rightarrow \mathbb{R}^{2m}$ allows us to re-order matrix multiplication by associativity, reducing prefill from $\Theta(n^2d)$ to $\mathcal{O}(nmd)$ time and decoding to $\mathcal{O}(md)$ constant memory.
2. **Session 2 — Coresets, Discrepancy, and Attention Compression (Sections 4–8):** Rather than approximating the kernel in feature space, can we select a small weighted subset of tokens (**a coreset**) to compress the KV-cache? We build coreset theory from first principles (1D quantiles, Hoeffding + ε -net sampling, and RFF-based Kernel Density Estimation), prove the **Halving Lemma** and the **Karnin & Liberty** [5] $\mathcal{O}(\sqrt{d}/\varepsilon)$ Gaussian KDE bound, and conclude with the **Liberty & Andoni** impossibility result showing why softmax normalization forces worst-case attention coresets to have size $\Omega(n)$.

Session 1: Attention Bottlenecks, The Kernel Trick, and Linearization

1 Attention Mechanics, Frontier Scale, and Baselines

1.1 Scaled Dot-Product Attention

Given an input sequence of n token embeddings $X \in \mathbb{R}^{n \times d_{\text{model}}}$, a single attention head projects X using learned weight matrices $W_Q, W_K, W_V \in \mathbb{R}^{d_{\text{model}} \times d}$ into queries, keys, and values:

$$Q = XW_Q, \quad K = XW_K, \quad V = XW_V \in \mathbb{R}^{n \times d}.$$

(Note: Modern LLMs inject sequence order at this step via **Rotary Position Embeddings (RoPE)** [9], applying 2D rotation blocks R_i to query q_i and R_j to key k_j so that their inner product $q_i^T R_{j-i} k_j$ depends only on relative token distance $i - j$.)

Definition 1.1 (Scaled Dot-Product Attention [10]). For queries $q_1, \dots, q_n \in \mathbb{R}^d$, keys $k_1, \dots, k_n \in \mathbb{R}^d$, and values $v_1, \dots, v_n \in \mathbb{R}^d$, the attention output for token $i \in [n]$ is the convex combination:

$$y_i = \sum_{j=1}^n P_{ij} v_j, \quad \text{where} \quad P_{ij} = \frac{\exp(q_i^T k_j / \sqrt{d})}{\sum_{l=1}^n \exp(q_i^T k_l / \sqrt{d})}. \quad (1)$$

In matrix form, with $S = \frac{1}{\sqrt{d}} Q K^T \in \mathbb{R}^{n \times n}$ and row-wise $P = \text{softmax}(S) \in \mathbb{R}^{n \times n}$, we have $Y = P V \in \mathbb{R}^{n \times d}$. (In autoregressive decoding, causal masking sets $S_{ij} = -\infty$ for $j > i$.)

Why Attention Works. While MLP layers apply static weights to each token in isolation, attention acts as a **dynamic, content-addressable associative memory**: query q_i searches against keys k_j to route information from values v_j on the fly.

1.2 Concrete Numbers: Prefill FLOPs vs. Generation KV-Cache

Consider a standard frontier model architecture (**Llama 3 70B** [4]):

- Layers: $N_{\text{layers}} = 80$
- Query heads: $H_Q = 64$, Key/Value heads: $H_{KV} = 8$ (Grouped-Query Attention), head dimension: $d = 128$
- Model dimension: $d_{\text{model}} = H_Q \cdot d = 8192$
- Context length: $n = 128,000$ tokens

Phase 1: Prefill (Prompt Ingestion — Compute-Bound). All n prompt tokens arrive simultaneously. Computing QK^T and PV across all heads and layers requires:

$$\text{FLOPs}_{\text{attn}} = 2 \cdot n^2 \cdot d \cdot H_Q \cdot N_{\text{layers}}.$$

- At $n = 4\text{k}$: $\text{FLOPs}_{\text{attn}} \approx 2 \times (4096)^2 \times 128 \times 64 \times 80 \approx \mathbf{22}$ TFLOPs.
- At $n = 128\text{k}$ ($32\times$ longer): n^2 grows by $32^2 = 1024\times$, requiring **22.5** PFLOPs (2.25×10^{16} FLOPs).
- Because MLP FLOPs scale linearly as $\mathcal{O}(n d_{\text{model}}^2)$ while attention scales quadratically as $\Theta(n^2 d_{\text{model}})$, attention dominates prefill compute at long contexts.

Phase 2: Autoregressive Generation (Token-by-Token — Memory-Bandwidth-Bound). To generate token t , we compute a single query $q_t \in \mathbb{R}^d$ and attend to all cached past keys $K_{\leq t}$ and values $V_{\leq t}$ (**the KV-cache**). In FP16 (2 bytes per number), the KV-cache size is:

$$\text{Memory}_{KV} = 2 \cdot N_{\text{layers}} \cdot H_{KV} \cdot d \cdot n \cdot 2 \text{ bytes}.$$

- Per token: $2 \times 80 \times 8 \times 128 \times 2 \approx \mathbf{328}$ KB / token.

- At $n = 128k$: $328 \text{ KB} \times 131,072 \approx \mathbf{43 \text{ GB}}$ for a **single sequence**.
- Generating just **one** token requires streaming all 43 GB from GPU High-Bandwidth Memory (HBM) to on-chip SRAM to perform only $\mathcal{O}(nd)$ arithmetic operations.

1.3 Hardware Acceleration: FlashAttention & Online Softmax

On an H100 GPU, HBM is large (80 GB) but relatively slow ($\sim 3.3 \text{ TB/s}$), whereas on-chip SRAM is tiny ($\sim 228 \text{ KB}$ per SM) but $10\times$ faster ($\sim 33 \text{ TB/s}$). Naive attention writes the $n \times n$ matrix S to HBM, incurring $\Theta(n^2)$ slow HBM reads/writes.

FlashAttention [3] never materializes S or P in HBM. It loads small tiles of Q, K, V into SRAM and updates the softmax output in a single streaming pass using **Online Softmax**:

$$m' = \max(m, s_j), \quad z' = z e^{m-m'} + e^{s_j-m'}, \quad y' = \frac{z e^{m-m'}}{z'} y + \frac{e^{s_j-m'}}{z'} v_j.$$

Systems vs. Algorithms Takeaway: FlashAttention reduces HBM IO traffic from $\Theta(n^2)$ to $\Theta(n^2 d / \text{SRAM})$, yielding a $3\text{--}5\times$ wall-clock speedup. However, **it still performs all $\Theta(n^2 d)$ arithmetic FLOPs**. Breaking the $\Theta(n^2)$ barrier requires **algorithmic approximations**.

1.4 ANN / MIPS for Attention

Because the softmax weight $P_{ij} \propto \exp(q_i^T k_j / \sqrt{d})$ grows exponentially with the inner product $q_i^T k_j$, attention scores are often concentrated on the keys most aligned with q_i . A natural algorithmic approach is to index the keys $k_1, \dots, k_n \in \mathbb{R}^d$ in an **Approximate Nearest Neighbor (ANN)** or **Maximum Inner Product Search (MIPS)** data structure (such as Locality-Sensitive Hashing in *Reformer* [6], or clustering/graph indices) and approximate row i using only the retrieved top- k keys $N_k(q_i) \subset [n]$:

$$\hat{y}_i = \sum_{j \in N_k(q_i)} \frac{\exp(q_i^T k_j / \sqrt{d})}{\sum_{l \in N_k(q_i)} \exp(q_i^T k_l / \sqrt{d})} v_j.$$

• Benefits:

1. *Sub-Quadratic Compute & Exact Lookup:* Reduces per-query complexity from $\Theta(nd)$ to sublinear search (e.g., $\mathcal{O}(kd)$ or $\mathcal{O}(n^\rho d)$), excelling when attention is sparse or peaked (such as long-range factual lookup).
2. *Selective Memory Access:* During decoding, only the candidate keys and values in $N_k(q_i)$ need to be fetched and aggregated rather than scanning the full context.

• Drawbacks & Trade-Offs:

1. *Diffuse Regimes & Normalizer Bias:* When attention is spread across many tokens ($P_{ij} \approx 1/n$), restricting to top- k omits tail mass and biases the normalizing denominator $Z_i = \sum_{j=1}^n \exp(q_i^T k_j / \sqrt{d})$.
2. *Dynamic Index Overhead & Hardware Irregularity:* In autoregressive decoding, new keys arrive at every token step, requiring online index updates; furthermore, sparse irregular lookups underutilize GPU Tensor Cores compared to dense matrix multiplications.

2 The Classical Kernel Trick & Random Fourier Features

2.1 Motivation from Kernel SVM: Primal vs. Dual & The $\mathcal{O}(N^2)$ Bottleneck

Why do kernels arise, and why do they create a quadratic bottleneck? Consider binary classification with a **Support Vector Machine (SVM)** on N training points $(x_1, y_1), \dots, (x_N, y_N) \in \mathbb{R}^d \times \{-1, +1\}$:

1. **Linear SVM (Primal Representation):** We learn a weight vector $w \in \mathbb{R}^d$ (e.g., minimizing $\frac{\lambda}{2} \|w\|_2^2 + \sum_{j=1}^N \max(0, 1 - y_j w^T x_j)$). By the Representer Theorem, the optimal weight vector is a linear combination of the data points:

$$w^* = \sum_{j=1}^N \alpha_j x_j \in \mathbb{R}^d, \quad \text{and prediction is} \quad f(q) = (w^*)^T q.$$

Because we can explicitly sum and store the single vector $w^* \in \mathbb{R}^d$, inference on a query q takes only $\mathcal{O}(d)$ **time** and $\mathcal{O}(d)$ **memory**—independent of N .

2. **Non-Linear Separation via Feature Maps:** When data is not linearly separable in $\mathbb{R}^d$, we map inputs via a non-linear feature map $\Phi : \mathbb{R}^d \rightarrow \mathcal{H}$ into a high- (or infinite-) dimensional Hilbert space $\mathcal{H}$ where the classes become linearly separable.
3. **The Kernel Trick (Dual Representation):** If $\dim(\mathcal{H})$ is huge or infinite, we **cannot** explicitly compute or store the primal vector $w^* = \sum_{j=1}^N \alpha_j \Phi(x_j) \in \mathcal{H}$. Instead, substituting w^* into $f(q) = \langle w^*, \Phi(q) \rangle_{\mathcal{H}}$ by linearity yields the **dual representation**:

$$f(q) = \left\langle \sum_{j=1}^N \alpha_j \Phi(x_j), \Phi(q) \right\rangle_{\mathcal{H}} = \sum_{j=1}^N \alpha_j \langle \Phi(x_j), \Phi(q) \rangle_{\mathcal{H}} = \sum_{j=1}^N \alpha_j \mathcal{K}(x_j, q). \quad (2)$$

Definition 2.1 (Mercer Kernel). *A symmetric function $\mathcal{K} : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ is a **positive semi-definite (Mercer) kernel** if there exists a Hilbert space $\mathcal{H}$ and a feature map $\Phi : \mathbb{R}^d \rightarrow \mathcal{H}$ such that:*

$$\mathcal{K}(x, z) = \langle \Phi(x), \Phi(z) \rangle_{\mathcal{H}} \quad \forall x, z \in \mathbb{R}^d.$$

Example (Quadratic Polynomial Kernel). For $\mathcal{K}(x, z) = (x^T z)^2$ on $\mathbb{R}^d$:

$$(x^T z)^2 = \left(\sum_{a=1}^d x_a z_a \right) \left(\sum_{b=1}^d x_b z_b \right) = \sum_{a,b=1}^d (x_a x_b)(z_a z_b) = \langle \Phi(x), \Phi(z) \rangle,$$

where $\Phi(x) = (x_a x_b)_{a,b=1}^d \in \mathbb{R}^{d^2}$. Evaluating $\mathcal{K}(x, z) = (x^T z)^2$ takes $\mathcal{O}(d)$ time without forming $\Phi(x) \in \mathbb{R}^{d^2}$.

The $\Theta(N^2)$ Bottleneck of the Kernel Trick (and the Exact Parallel to Attention). Because Kernel SVM operates in the dual (2), it must retain all N support vectors in memory, pay $\Theta(Nd)$ per query, and compute the $N \times N$ Gram matrix $K_{ij} = \mathcal{K}(x_i, x_j)$ in $\Theta(N^2 d)$ time during training. Compare

(2) to Attention (1):

$$\text{Kernel SVM: } f(q) = \sum_{j=1}^N \alpha_j \mathcal{K}(q, x_j) \quad \longleftrightarrow \quad \text{Attention: } y(q) = \frac{1}{Z(q)} \sum_{j=1}^n v_j \mathcal{K}(q, k_j).$$

Softmax Attention is a Kernel SVM in dual form—where the keys $k_1, \dots, k_n$ are the support vectors (the KV-cache) and the values $v_1, \dots, v_n$ are the dual weights α_j .

2.2 Inverting the Kernel Trick: Random Fourier Features (Rahimi & Recht, 2007)

To escape the $\Theta(N^2)$ dual bottleneck, **Rahimi and Recht** [8] proposed **inverting the kernel trick**—moving from the dual (2) **back to the primal** by constructing an explicit randomized feature map for the **Gaussian (RBF) Kernel** $\mathcal{K}_{\text{gauss}}(x, y) = \exp(-\|x - y\|_2^2/2)$.

Draw a random frequency vector $\omega \sim \mathcal{N}(0, I_d)$ and define the complex randomized feature:

$$\phi(x) \triangleq e^{i\omega^T x} \in \mathbb{C}. \quad (3)$$

Theorem 2.2 (Unbiasedness of the Complex Fourier Feature [8]). *For any $x, y \in \mathbb{R}^d$ and $\omega \sim \mathcal{N}(0, I_d)$:*

$$\mathbb{E}_{\omega \sim \mathcal{N}(0, I_d)} [\phi(x) \phi(y)^*] = \exp\left(-\frac{\|x - y\|_2^2}{2}\right) = \mathcal{K}_{\text{gauss}}(x, y). \quad (4)$$

Proof. We establish the identity in two elementary steps:

1. **1D Gaussian Moment Generating Function (MGF):** For a scalar $z \sim \mathcal{N}(0, 1)$ with density $\psi(z) = \frac{1}{\sqrt{2\pi}} e^{-z^2/2}$ and any scalar $t \in \mathbb{C}$, completing the square gives $-\frac{1}{2}z^2 + tz = -\frac{1}{2}(z - t)^2 + \frac{1}{2}t^2$, so:

$$\mathbb{E}_{z \sim \mathcal{N}(0, 1)} [e^{tz}] = \int_{-\infty}^{\infty} e^{tz} \psi(z) dz = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-\frac{1}{2}z^2 + tz} dz = e^{t^2/2} \underbrace{\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-\frac{1}{2}(z-t)^2} dz}_{=1} = e^{t^2/2}.$$

2. Rotational Invariance (2-Stability) of Gaussians:

- If $x \sim \mathcal{N}(0, 1)$, then $\sigma \cdot x \sim \mathcal{N}(0, \sigma^2)$.
- For two independent Gaussians $x_1 \sim \mathcal{N}(0, \sigma_1^2)$ and $x_2 \sim \mathcal{N}(0, \sigma_2^2)$, $x_1 + x_2 \sim \mathcal{N}(0, \sigma_1^2 + \sigma_2^2)$.

Consequently, for $\omega \sim \mathcal{N}(0, I_d)$, the inner product $\omega^T(x - y) = \sum_{j=1}^d \omega_j(x_j - y_j)$ is distributed as $z \|x - y\|_2$, where $z \sim \mathcal{N}(0, 1)$.

Substituting $\phi(x) \phi(y)^* = e^{i\omega^T(x-y)} \stackrel{d}{=} e^{(i\|x-y\|_2)z}$ and applying the 1D MGF with $t = i\|x - y\|_2$ (so $t^2 = -\|x - y\|_2^2$) yields:

$$\mathbb{E}_{\omega \sim \mathcal{N}(0, I_d)} [\phi(x) \phi(y)^*] = \mathbb{E}_{z \sim \mathcal{N}(0, 1)} [e^{(i\|x-y\|_2)z}] = \exp\left(\frac{(i\|x - y\|_2)^2}{2}\right) = \exp\left(-\frac{\|x - y\|_2^2}{2}\right). \quad \square$$

2.3 Explicit $2m$ -Dimensional Feature Map & Variance Bound

Because $\mathcal{K}_{\text{gauss}}(x, y)$ is real, we only need the real part of $\phi(x)\phi(y)^*$:

$$\text{Re}(e^{i\omega^T x} e^{-i\omega^T y}) = \cos(\omega^T(x - y)) = \cos(\omega^T x) \cos(\omega^T y) + \sin(\omega^T x) \sin(\omega^T y).$$

Drawing m independent vectors $\omega_1, \dots, \omega_m \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_d)$, we obtain the explicit real feature map $\phi : \mathbb{R}^d \rightarrow \mathbb{R}^{2m}$:

$$\phi(x) \triangleq \frac{1}{\sqrt{m}} \left[\cos(\omega_1^T x), \sin(\omega_1^T x), \dots, \cos(\omega_m^T x), \sin(\omega_m^T x) \right]^T \in \mathbb{R}^{2m}. \quad (5)$$

(Equivalently, in complex coordinates, $\phi(x) = \frac{1}{\sqrt{m}} [e^{i\omega_1^T x}, \dots, e^{i\omega_m^T x}]^T \in \mathbb{C}^m$.)

Lemma 2.3 (Unbiasedness, Bounded Variance & Concentration of RFF). *For any $x, y \in \mathbb{R}^d$, the $2m$ -dimensional RFF inner product $\phi(x)^T \phi(y)$ satisfies:*

1. **Unbiasedness:** $\mathbb{E}[\phi(x)^T \phi(y)] = \exp(-\|x - y\|_2^2/2)$.
2. **Bounded Variance:** $\text{Var}(\phi(x)^T \phi(y)) \leq \frac{1}{m}$ (independent of dimension d and norms $\|x\|, \|y\|$).
3. **Concentration:** For $m \geq \frac{2}{\varepsilon^2} \ln \frac{2}{\delta}$, $\Pr[|\phi(x)^T \phi(y) - \mathcal{K}_{\text{gauss}}(x, y)| \leq \varepsilon] \geq 1 - \delta$.

Proof. For each $r \in [m]$, let $Z_r \triangleq \text{Re}(e^{i\omega_r^T x} e^{-i\omega_r^T y}) = \cos(\omega_r^T(x - y))$, so $\phi(x)^T \phi(y) = \frac{1}{m} \sum_{r=1}^m Z_r$.

1. **Unbiasedness:** By Theorem 2.2, $\mathbb{E}[Z_r] = \text{Re}(\mathbb{E}[e^{i\omega_r^T(x-y)}]) = \mathcal{K}_{\text{gauss}}(x, y)$.
2. **Variance:** Since $|Z_r| \leq 1$ (or $|e^{i\omega_r^T(x-y)}| = 1$), $\text{Var}(Z_r) \leq \mathbb{E}[Z_r^2] \leq 1$. By independence, $\text{Var}(\frac{1}{m} \sum_{r=1}^m Z_r) \leq \frac{1}{m}$.
3. **Concentration:** Since $Z_1, \dots, Z_m \in [-1, 1]$ are i.i.d. bounded variables of range 2, Hoeffding's inequality directly gives $\Pr[|\frac{1}{m} \sum_{r=1}^m Z_r - \mathcal{K}_{\text{gauss}}(x, y)| \geq \varepsilon] \leq 2 \exp(-m\varepsilon^2/2) \leq \delta$ for $m \geq \frac{2}{\varepsilon^2} \ln \frac{2}{\delta}$. $\square$

3 Linearizing Attention via Random Fourier Features

3.1 Reduction from Dot-Product Softmax to the Gaussian Kernel

To apply RFF to attention, we express the softmax dot-product kernel $\exp(q^T k / \sqrt{d})$ in terms of the shift-invariant Gaussian kernel. Rescale $\tilde{q} \triangleq q/d^{1/4}$ and $\tilde{k} \triangleq k/d^{1/4}$ so that $q^T k / \sqrt{d} = \tilde{q}^T \tilde{k}$.

Using the polarization identity $\|\tilde{q} - \tilde{k}\|_2^2 = \|\tilde{q}\|_2^2 + \|\tilde{k}\|_2^2 - 2\tilde{q}^T \tilde{k}$, we have:

$$\exp(\tilde{q}^T \tilde{k}) = \exp\left(\frac{\|\tilde{q}\|_2^2}{2}\right) \cdot \exp\left(-\frac{\|\tilde{q} - \tilde{k}\|_2^2}{2}\right) \cdot \exp\left(\frac{\|\tilde{k}\|_2^2}{2}\right). \quad (6)$$

Define the rescaled feature map $\psi : \mathbb{R}^d \rightarrow \mathbb{R}^{2m}$ by absorbing the scalar norm multiplier into ϕ :

$$\psi(x) \triangleq \exp\left(\frac{\|x\|_2^2}{2}\right) \phi(x) \in \mathbb{R}^{2m}. \quad (7)$$

Then by Lemma 2.3 and (6), $\psi(\tilde{q})^T \psi(\tilde{k})$ is an unbiased estimator of the softmax kernel:

$$\mathbb{E}[\psi(\tilde{q})^T \psi(\tilde{k})] = \exp\left(\frac{q^T k}{\sqrt{d}}\right).$$

(Remark: Notice that the query scalar $\exp(\|\tilde{q}_i\|_2^2/2)$ appears in **both** the numerator and denominator of row i 's softmax (1) and cancels out identically.)

3.2 The Associativity Trick: $\mathcal{O}(n^2d) \rightarrow \mathcal{O}(nmd)$

Replace $\exp(q_i^T k_j / \sqrt{d})$ in the attention equation (1) with the linear inner product $\psi(\tilde{q}_i)^T \psi(\tilde{k}_j)$:

$$\hat{y}_i = \frac{\sum_{j=1}^n (\psi(\tilde{q}_i)^T \psi(\tilde{k}_j)) v_j}{\sum_{j=1}^n \psi(\tilde{q}_i)^T \psi(\tilde{k}_j)}.$$

By **associativity of matrix multiplication**, pull $\psi(\tilde{q}_i)^T$ outside the summation over j :

$$\hat{y}_i = \frac{\psi(\tilde{q}_i)^T \left(\sum_{j=1}^n \psi(\tilde{k}_j) v_j^T \right)}{\psi(\tilde{q}_i)^T \left(\sum_{j=1}^n \psi(\tilde{k}_j) \right)} = \frac{\psi(\tilde{q}_i)^T M}{\psi(\tilde{q}_i)^T z}, \quad (8)$$

where the summary statistics M and z are computed **once** across all n keys and values:

$$M \triangleq \sum_{j=1}^n \psi(\tilde{k}_j) v_j^T \in \mathbb{R}^{2m \times d}, \quad z \triangleq \sum_{j=1}^n \psi(\tilde{k}_j) \in \mathbb{R}^{2m}. \quad (9)$$

Algorithm LINEAR-ATTENTION($Q, K, V \in \mathbb{R}^{n \times d}$, feature dimension m)

1. Sample $\omega_1, \dots, \omega_m \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_d)$.
2. Compute feature maps $\psi(\tilde{q}_i), \psi(\tilde{k}_j) \in \mathbb{R}^{2m}$ for all $i, j \in [n]$. // $\mathcal{O}(nmd)$
3. Aggregate the KV summary matrix and normalizer:
$$M \leftarrow \sum_{j=1}^n \psi(\tilde{k}_j) v_j^T \in \mathbb{R}^{2m \times d}, \quad z \leftarrow \sum_{j=1}^n \psi(\tilde{k}_j) \in \mathbb{R}^{2m}. \quad // \mathcal{O}(nmd)$$
4. **For each** query $i = 1, \dots, n$: compute $\hat{y}_i \leftarrow \frac{\psi(\tilde{q}_i)^T M}{\psi(\tilde{q}_i)^T z}$. // $\mathcal{O}(nmd)$
5. **Return** $\hat{Y} = [\hat{y}_1; \dots; \hat{y}_n] \in \mathbb{R}^{n \times d}$.

- **Prefill Complexity:** Drops from $\Theta(n^2d)$ to $\mathcal{O}(nmd)$ — **strictly linear in sequence length n .**
- **Autoregressive Decoding (Causal Linear RNN):** With causal masking ($j \leq t$), we maintain running state (M_t, z_t) updated in $\mathcal{O}(md)$ **constant time and constant memory per step:**

$$M_t = M_{t-1} + \psi(\tilde{k}_t) v_t^T, \quad z_t = z_{t-1} + \psi(\tilde{k}_t), \quad \hat{y}_t = \frac{\psi(\tilde{q}_t)^T M_t}{\psi(\tilde{q}_t)^T z_t}. \quad (10)$$

The $\mathcal{O}(nd)$ KV-cache is completely eliminated—replaced by the fixed-size state $M_t \in \mathbb{R}^{2m \times d}$.

Remark 3.1 (Expressivity Limits & Suggested Reading).

1. **The “Needle in a Haystack” Barrier:** Compressing n key-value pairs into a fixed matrix $M \in \mathbb{R}^{2m \times d}$ of rank at most $\min(2m, d)$ imposes a hard information capacity ceiling. When $n \gg m$, linear attention cannot perform exact token retrieval (“needle in a haystack”) without cross-talk interference.
2. **Reading Note on Positive Features:** Choromanski et al. (*Performers* / *FAVOR+* [2]) proposed positive exponential features $\phi_{\text{pos}}(x) \propto \exp(\omega^T x)$ to avoid negative cosine estimates, though trigonometric RFF (5) has strictly bounded variance ($\leq 1/m$, Lemma 2.3) whereas exponential features suffer from high variance as $\|x\|$ grows.

Session 2: Coresets, Discrepancy, and Attention Compression

4 General Definition of Coresets & 1D Quantiles Warm-Up

Instead of approximating the kernel in feature space, can we compress the KV-cache by keeping only a **small subset of tokens**? This is the domain of **coresets**.

Definition 4.1 (ε -Coreset). *Given a dataset $X = \{x_1, \dots, x_n\} \subset \mathcal{X}$ and a query family $\mathcal{F} = \{f_q : \mathcal{X} \rightarrow [0, 1] \mid q \in \mathcal{Q}\}$, an ε -coreset is a subset $S \subseteq X$ with weights $w : S \rightarrow \mathbb{R}_{\geq 0}$ such that:*

$$\sup_{q \in \mathcal{Q}} \left| \sum_{x \in S} w(x) f_q(x) - \frac{1}{n} \sum_{i=1}^n f_q(x_i) \right| \leq \varepsilon. \quad (11)$$

When $w(x) = \frac{1}{|S|}$ for all $x \in S$, we call S an **unweighted coreset**.

4.1 Warm-Up Example: Deterministic Coreset for 1D Quantiles (Empirical CDF)

Let $X = \{x_1 \leq x_2 \leq \dots \leq x_n\} \subset \mathbb{R}$ be sorted scalars, and consider threshold queries $f_q(x) = \mathbb{I}[x \leq q]$ for $q \in \mathbb{R}$. Then $F(X, q) = \frac{1}{n} \sum_{i=1}^n \mathbb{I}[x_i \leq q]$ is the **empirical CDF (quantile rank)** of q .

Theorem 4.2 (Optimal 1D Quantile Coreset). *Let $m = \lceil 1/\varepsilon \rceil$ (assuming m divides n for simplicity), and let $S = \{x_{j \cdot (n/m)} : j = 1, 2, \dots, m\}$ consist of every (n/m) -th point in sorted order with uniform weight $1/m$. Then S is an unweighted ε -coreset of size $m = \mathcal{O}(1/\varepsilon)$:*

$$\sup_{q \in \mathbb{R}} \left| \frac{1}{m} \sum_{x \in S} \mathbb{I}[x \leq q] - \frac{1}{n} \sum_{i=1}^n \mathbb{I}[x_i \leq q] \right| \leq \varepsilon.$$

Proof. Partition the sorted array $x_1 \leq \dots \leq x_n$ into m contiguous blocks $B_1, \dots, B_m$, each containing $b = n/m \leq \varepsilon n$ elements. The coreset $S = \{s_1, \dots, s_m\}$ selects the **largest** element $s_j = x_{jb}$ of each block B_j .

Fix any query $q \in \mathbb{R}$, and let B_{j^*} be the unique block straddling q (so $s_{j^*-1} \leq q < s_{j^*}$):

1. **Blocks $j < j^*$:** Every element $x \in B_j$ satisfies $x \leq s_j \leq q$. Thus, both the dataset X and the coreset S count 100% of the first $j^* - 1$ blocks, contributing $\frac{j^*-1}{m}$ to both sums.

2. **Blocks $j > j^*$:** Every element $x \in B_j$ satisfies $x > q$, contributing 0 to both sums.

3. **The Straddled Block B_{j^*} :** In B_{j^*} , the coreset point $s_{j^*} > q$ contributes 0 to $F(S, q)$, whereas between 0 and b points of B_{j^*} satisfy $x_i \leq q$, contributing between 0 and $\frac{b}{n} = \frac{1}{m}$ to $F(X, q)$.

Hence, for every $q \in \mathbb{R}$, $0 \leq F(X, q) - F(S, q) \leq \frac{1}{m} \leq \varepsilon$. $\square$

5 Uniform Sampling Bounds via Hoeffding & ε -Nets

In d dimensions, we cannot sort the dataset. What coreset size does **Uniform Random Sampling** achieve?

Theorem 5.1 (Uniform Sampling Coreset Bound). *Let $\mathcal{F} = \{f_q : \mathcal{X} \rightarrow [0, 1] \mid q \in \mathcal{Q}\}$ be a family of queries over the unit ball $\mathcal{Q} = \{q \in \mathbb{R}^d : \|q\|_2 \leq 1\}$ that is L -Lipschitz in q (i.e., $|f_q(x) - f_{q'}(x)| \leq L\|q - q'\|_2$). Drawing m points $s_1, \dots, s_m \stackrel{i.i.d.}{\sim} \text{Unif}(X)$ yields an unweighted ε -coreset with probability at least $1 - \delta$ for:*

$$m = \mathcal{O}\left(\frac{d \log(L/\varepsilon) + \log(1/\delta)}{\varepsilon^2}\right). \quad (12)$$

Proof. Step 1 (Pointwise Concentration): For any fixed query $q \in \mathcal{Q}$, $Z_r = f_q(s_r) \in [0, 1]$ are i.i.d. with mean $F(X, q) = \frac{1}{n} \sum_{i=1}^n f_q(x_i)$. By Hoeffding's inequality:

$$\Pr\left[|F(S, q) - F(X, q)| \geq \frac{\varepsilon}{2}\right] \leq 2 \exp\left(-\frac{m\varepsilon^2}{2}\right).$$

Step 2 (ε -Net Covering of $\mathcal{Q}$): Let $\mathcal{N} \subset \mathcal{Q}$ be a minimal $(\frac{\varepsilon}{4L})$ -net of the unit ball $\mathcal{Q}$: for every $q \in \mathcal{Q}$, there exists $\hat{q} \in \mathcal{N}$ with $\|q - \hat{q}\|_2 \leq \frac{\varepsilon}{4L}$. By a standard volume packing argument, $|\mathcal{N}| \leq \left(\frac{12L}{\varepsilon}\right)^d$.

Step 3 (Union Bound & Triangle Inequality): By a union bound over all $\hat{q} \in \mathcal{N}$, the probability that $|F(S, \hat{q}) - F(X, \hat{q})| \geq \varepsilon/2$ for *any* net point $\hat{q} \in \mathcal{N}$ is at most:

$$|\mathcal{N}| \cdot 2 \exp\left(-\frac{m\varepsilon^2}{2}\right) \leq \left(\frac{12L}{\varepsilon}\right)^d \cdot 2 \exp\left(-\frac{m\varepsilon^2}{2}\right) \leq \delta,$$

which holds for $m = \frac{2}{\varepsilon^2} \left(d \ln \frac{12L}{\varepsilon} + \ln \frac{2}{\delta}\right)$. Whenever this event holds, for *any* $q \in \mathcal{Q}$ with nearest net point $\hat{q} \in \mathcal{N}$, L -Lipschitzness gives $|F(S, q) - F(S, \hat{q})| \leq \frac{\varepsilon}{4}$ and $|F(X, q) - F(X, \hat{q})| \leq \frac{\varepsilon}{4}$. By the triangle inequality:

$$|F(S, q) - F(X, q)| \leq |F(S, q) - F(S, \hat{q})| + |F(S, \hat{q}) - F(X, \hat{q})| + |F(X, \hat{q}) - F(X, q)| \leq \frac{\varepsilon}{4} + \frac{\varepsilon}{2} + \frac{\varepsilon}{4} = \varepsilon. \quad \square$$

6 Coresets for Kernel Density Estimation via RFF

Given $X = \{x_1, \dots, x_n\} \subset \mathbb{R}^d$, the **Gaussian Kernel Density Estimate (KDE)** at query $q \in \mathbb{R}^d$ is:

$$F(X, q) \triangleq \frac{1}{n} \sum_{i=1}^n \mathcal{K}(q, x_i) = \frac{1}{n} \sum_{i=1}^n \exp\left(-\frac{\|q - x_i\|_2^2}{2}\right). \quad (13)$$

Notice that the ε -net bound (Theorem 5.1) gives $m = \mathcal{O}(d/\varepsilon^2)$ and only holds for bounded $\|q\| \leq 1$. **Using our linearization / feature map from Session 1, we can prove a dimension-free $m = \mathcal{O}(1/\varepsilon^2)$ bound over all $q \in \mathbb{R}^d$ in four lines.**

Theorem 6.1 (Dimension-Free KDE Coreset via Kernel Feature Maps). *For any dataset $X = \{x_1, \dots, x_n\} \subset \mathbb{R}^d$, a uniform random sample $S = \{s_1, \dots, s_m\} \stackrel{\text{i.i.d.}}{\sim} \text{Unif}(X)$ of size $m = \frac{1}{\varepsilon^2}$ satisfies:*

$$\mathbb{E}_S \left[\sup_{q \in \mathbb{R}^d} \left| \frac{1}{m} \sum_{r=1}^m \mathcal{K}(q, s_r) - \frac{1}{n} \sum_{i=1}^n \mathcal{K}(q, x_i) \right| \right] \leq \varepsilon.$$

Proof. Step 1 (Linearization in Feature Space): By Definition 2.1 (or the continuous RFF representation of Theorem 2.2), write $\mathcal{K}(q, x) = \langle \Phi(q), \Phi(x) \rangle_{\mathcal{H}}$, where every point has unit feature norm: $\|\Phi(x)\|_{\mathcal{H}} = \sqrt{\mathcal{K}(x, x)} = 1$.

Step 2 (Factoring Out q via Cauchy-Schwarz): Let $\mu_X \triangleq \frac{1}{n} \sum_{i=1}^n \Phi(x_i)$ and $\mu_S \triangleq \frac{1}{m} \sum_{r=1}^m \Phi(s_r)$. By linearity of inner product and the Cauchy-Schwarz inequality:

$$\sup_{q \in \mathbb{R}^d} |F(S, q) - F(X, q)| = \sup_{q \in \mathbb{R}^d} |\langle \Phi(q), \mu_S - \mu_X \rangle_{\mathcal{H}}| \leq \sup_{q \in \mathbb{R}^d} \underbrace{\|\Phi(q)\|_{\mathcal{H}}}_{=1} \cdot \|\mu_S - \mu_X\|_{\mathcal{H}} = \|\mu_S - \mu_X\|_{\mathcal{H}}. \quad (14)$$

Step 3 (Expected Squared Norm): Because $s_1, \dots, s_m \stackrel{\text{i.i.d.}}{\sim} \text{Unif}(X)$ have mean $\mathbb{E}[\Phi(s_r)] = \mu_X$, cross terms vanish in expectation:

$$\mathbb{E}[\|\mu_S - \mu_X\|_{\mathcal{H}}^2] = \frac{1}{m^2} \sum_{r=1}^m \mathbb{E}[\|\Phi(s_r) - \mu_X\|_{\mathcal{H}}^2] \leq \frac{1}{m} \mathbb{E}[\|\Phi(s_1)\|_{\mathcal{H}}^2] = \frac{1}{m}.$$

By Jensen's inequality, $\mathbb{E}[\sup_{q \in \mathbb{R}^d} |F(S, q) - F(X, q)|] \leq \sqrt{\mathbb{E}[\|\mu_S - \mu_X\|_{\mathcal{H}}^2]} \leq \frac{1}{\sqrt{m}} = \varepsilon. \quad \square$

7 Breaking the $1/\varepsilon^2$ Barrier via Discrepancy (Karnin & Liberty, 2019)

Uniform sampling requires $m = \mathcal{O}(1/\varepsilon^2)$ because random i.i.d. sampling accumulates error like a random walk ($\pm\sqrt{n}$ over n points). Can we actively **choose** signs $\sigma \in \{-1, +1\}^n$ so errors cancel out?

Definition 7.1 (Class Discrepancy — Karnin & Liberty, FOCS 2019 [5]). *The **class discrepancy** of a function family $\mathcal{F}$ on n -point datasets is:*

$$D_n(\mathcal{F}) \triangleq \max_{X: |X|=n} \min_{\substack{\sigma \in \{-1, +1\}^n \\ \sum_i \sigma_i = 0}} \sup_{f \in \mathcal{F}} \frac{1}{n} \left| \sum_{i=1}^n \sigma_i f(x_i) \right|. \quad (15)$$

7.1 The Iterated Halving Reduction: From Discrepancy to Coresets

Lemma 7.2 (Halving Lemma — Karnin & Liberty, 2019 [5]). *If a function family $\mathcal{F}$ has discrepancy $D_n(\mathcal{F}) \leq \frac{c}{n}$, then any dataset X of size n admits an unweighted ε -coreset $S \subseteq X$ of size:*

$$m = \mathcal{O}\left(\frac{c}{\varepsilon}\right).$$

Proof. Step 1 (Single Halving Step): Given $X_0 = X$ of size n (assume n is a power of 2), let $\sigma \in \{-1, +1\}^n$ with $\sum_{i=1}^n \sigma_i = 0$ be the optimal sign assignment achieving (15), and let $X_1 = \{x_i \in X_0 : \sigma_i = +1\}$ be the positive half of size $n/2$. Since $\frac{1}{n/2} \mathbb{I}[\sigma_i = +1] - \frac{1}{n} = \frac{\sigma_i}{n}$, for every $f \in \mathcal{F}$:

$$|F(X_1, f) - F(X_0, f)| = \left| \frac{2}{n} \sum_{x_i \in X_1} f(x_i) - \frac{1}{n} \sum_{i=1}^n f(x_i) \right| = \frac{1}{n} \left| \sum_{i=1}^n \sigma_i f(x_i) \right| \leq D_n(\mathcal{F}) \leq \frac{c}{n}.$$

Step 2 (Telescoping T Halving Steps): Recursively halve the surviving set T times: $X_0 \rightarrow X_1 \rightarrow \dots \rightarrow X_T$, where $|X_t| = n/2^t$ and the final coreset $S = X_T$ has size $m = n/2^T$. By the triangle inequality, the total error is a geometric series dominated by the final step:

$$\sup_{f \in \mathcal{F}} |F(S, f) - F(X, f)| \leq \sum_{t=0}^{T-1} \sup_{f \in \mathcal{F}} |F(X_{t+1}, f) - F(X_t, f)| \leq \sum_{t=0}^{T-1} \frac{c}{n/2^t} = \frac{c}{n} \sum_{t=0}^{T-1} 2^t \leq \frac{2c}{n/2^T} = \frac{2c}{m}.$$

Setting $\frac{2c}{m} \leq \varepsilon$ yields a coreset of size $m = \frac{2c}{\varepsilon} = \mathcal{O}(c/\varepsilon)$. $\square$

7.2 Kernel Discrepancy & The Karnin–Liberty KDE Bound

First, notice how easily we can match the $\sqrt{n}$ bound deterministically in our kernel feature space $\mathcal{H}$ (where $\|\Phi(x)\|_{\mathcal{H}} \leq 1$):

Proposition 7.3 (Greedy Kernel Halving). *For any vectors $\Phi(x_1), \dots, \Phi(x_n) \in \mathcal{H}$ with $\|\Phi(x_i)\|_{\mathcal{H}} \leq 1$, choosing signs $\sigma_i \in \{-1, +1\}$ greedily so that $\sigma_i \langle \sum_{j < i} \sigma_j \Phi(x_j), \Phi(x_i) \rangle_{\mathcal{H}} \leq 0$ guarantees:*

$$\left\| \sum_{i=1}^n \sigma_i \Phi(x_i) \right\|_{\mathcal{H}} \leq \sqrt{n}.$$

Proof. Let $v_i = \sum_{j=1}^i \sigma_j \Phi(x_j)$. At step i , $\|v_i\|_{\mathcal{H}}^2 = \|v_{i-1}\|_{\mathcal{H}}^2 + \|\Phi(x_i)\|_{\mathcal{H}}^2 + 2\sigma_i \langle v_{i-1}, \Phi(x_i) \rangle_{\mathcal{H}} \leq \|v_{i-1}\|_{\mathcal{H}}^2 + 1$. By induction, $\|v_n\|_{\mathcal{H}}^2 \leq n$. $\square$

Even more powerfully, by applying **Banaszczyk’s Vector Balancing Theorem** [1] to the Taylor expansion of analytic functions of dot products in $\mathbb{R}^d$, Karnin and Liberty proved a $1/n$ discrepancy rate:

Theorem 7.4 (Karnin & Liberty, FOCS 2019 [5]). *For the Gaussian kernel $\mathcal{K}(q, x) = \exp(-\|q - x\|_2^2/2)$ (and any bounded analytic function of dot products in $\mathbb{R}^d$), the class discrepancy satisfies:*

$$D_n(\mathcal{F}) = \mathcal{O}\left(\frac{\sqrt{d}}{n}\right).$$

Consequently, by the Halving Lemma (Lemma 7.2), Gaussian KDE admits an unweighted ε -coreset of size:

$$m = \mathcal{O}\left(\frac{\sqrt{d}}{\varepsilon}\right). \quad (16)$$

(This resolved the long-standing open problem on the coreset complexity of Gaussian KDE, matching the $\Omega(\sqrt{d}/\varepsilon)$ lower bound of Phillips & Tai [7].)

8 Outcome for Attention (Liberty & Andoni) & Practical Eviction

8.1 Why Softmax Attention Breaks KDE Coresets: The Normalization Barrier

Why doesn't Theorem 7.4 immediately give an $\mathcal{O}(\sqrt{d}/\varepsilon)$ -sized KV-cache coreset for LLM attention?

- **Gaussian KDE** approximates an **unnormalized average** $Z(q) = \frac{1}{n} \sum_{j=1}^n \mathcal{K}(q, k_j)$ up to **additive error** $\pm \varepsilon$.
- **Softmax Attention** computes a **normalized ratio**:

$$A(q) = \frac{\frac{1}{n} \sum_{j=1}^n \exp(q^T k_j / \sqrt{d}) v_j}{\frac{1}{n} \sum_{j=1}^n \exp(q^T k_j / \sqrt{d})}.$$

When $Z(q) \ll \varepsilon$ (or when a single key dominates the sum while $n - 1$ keys have small inner product), an additive ε -approximation to numerator and denominator causes unbounded relative error in the ratio.

Theorem 8.1 (Liberty & Andoni — Attention Coreset Lower Bound). *Let $K = \{k_1, \dots, k_n\} \subset \mathbb{R}^d$ and $V = \{v_1, \dots, v_n\} \subset \mathbb{R}^d$ with $d \geq \Theta(\log n)$. For queries of norm $\|q\|_2 \leq \mathcal{O}(\sqrt{d} \log n)$, any coreset $S \subset [n]$ guaranteeing worst-case error $\|A(q) - A_S(q)\|_2 \leq \varepsilon$ for $\varepsilon < 1/\sqrt{2}$ **must retain all keys**:*

$$|S| = \Omega(n). \tag{17}$$

Geometric Intuition.

1. Let keys $k_1, \dots, k_n$ be mutually orthogonal unit vectors, paired with orthonormal values $v_1, \dots, v_n$.
2. If a candidate coreset S omits even a single key $i^* \notin S$, an adversary issues query $q = \beta k_{i^*}$ with $\beta = \Theta(\sqrt{d} \log n)$.
3. In the true attention $A(q)$, key i^* receives score $e^{\beta/\sqrt{d}} \gg n$, capturing $\approx 100\%$ of the softmax weight so $A(q) \approx v_{i^*}$. In the coreset $A_S(q)$, since $i^* \notin S$, the output lies in $\text{span}(\{v_j : j \in S\}) \perp v_{i^*}$, incurring $\Omega(1)$ Euclidean error.

8.2 Bridging Theory to Practice: H2O & Attention Sinks

Because worst-case sublinear attention coresets are impossible (Theorem 8.1), practical KV-cache compression relies on structural properties of **natural language queries**:

1. **Heavy Hitter Oracle (H2O; Zhang et al., NeurIPS 2023 [12])**: Empirically, LLM attention maps follow a power-law distribution where a small set of tokens (“**Heavy Hitters**”) repeatedly receive high attention across decoding steps. H2O maintains a cache of size $B \ll n$ by tracking cumulative attention scores $a_j = \sum_t P_{t,j}$ and greedily evicting the token with the smallest a_j .
2. **Attention Sinks (StreamingLLM; Xiao et al., ICLR 2024 [11])**: Because softmax probabilities must sum to 1, when a query has no relevant match in the context, the LLM dumps residual probability mass onto the first 1–4 tokens of the prompt (**attention sinks**). Pinning the first 4 tokens alongside a local sliding window stabilizes the softmax denominator $Z(q)$ during streaming inference.

Paradigm / Result	Mathematical Mechanism	Prefill FLOPs	KV-Cache Size	Guarantee & Key Takeaway
FlashAttention [3]	SRAM tiling + Online Softmax	$\Theta(n^2d)$	$\Theta(nd)$	Exact (0 error). Reduces HBM IO, but arithmetic FLOPs remain $\Theta(n^2d)$.
ANN / MIPS for Attention [6]	LSH / HNSW / IVF retrieval over keys	$\mathcal{O}(n \log n \cdot d)$	$\Theta(nd)$ (host/HBM)	Fast & natural for sparse/peaked attention; index overhead & diffuse-tail trade-offs.
RFF Linearization [8]	$\mathcal{K}(q, k) \approx \psi(q)^T \psi(k)$, associativity $\psi(q)^T M$	$\mathcal{O}(nmd)$	$\mathcal{O}(md)$ (constant in n)	Unbiased, $\text{Var} \leq 1/m$. Fixed state $M \in \mathbb{R}^{2m \times d}$ limits exact “needle in a haystack” recall.
1D Quantile Coreset	Equi-spaced sorted grid	—	$\mathcal{O}(1/\varepsilon)$	Deterministic $\pm\varepsilon$ CDF; relies on 1D total ordering.
Uniform Sampling KDE	i.i.d. sample + Cauchy-Schwarz in $\mathcal{H}$	—	$\mathcal{O}(1/\varepsilon^2)$	Dimension-free $\pm\varepsilon$ KDE; random-walk error prevents $1/\varepsilon$ scaling.
Discrepancy KDE Coreset [5]	Iterated halving + Banaszczyk balancing	—	$\mathcal{O}(\sqrt{d}/\varepsilon)$	Optimal $\pm\varepsilon$ Gaussian KDE; applies to unnormalized KDE.
Attention Coreset Lower Bound	Orthonormal keys/values + targeted query	$\Omega(n^2d)$	$\Omega(nd)$	$\Omega(n)$ keys required for $\varepsilon < 1/\sqrt{2}$. Worst-case sublinear attention coreset is impossible; motivates H2O / Sinks.

Table 1: Summary comparison of hardware, linearization, and coreset paradigms for LLM attention.

References

- [1] Wojciech Banaszczyk. Balancing vectors and Gaussian measures of n -dimensional convex bodies. *Random Structures & Algorithms*, 12(4):351–360, 1998.
- [2] Krzysztof Choromanski, Valerii Likhoshesterov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy Colwell, and Adrian Weller. Rethinking attention with Performers. In *International Conference on Learning Representations (ICLR)*, 2021.
- [3] 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)*, volume 35, pages 16344–16359, 2022.
- [4] Aaron Grattafiori et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

- [5] Zohar Karnin and Edo Liberty. Discrepancy, coresets, and sketches in machine learning. In *Proceedings of the 60th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 362–376, 2019. (*arXiv:1906.04845*).
- [6] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *International Conference on Learning Representations (ICLR)*, 2020.
- [7] Jeff M. Phillips and Wai Ming Tai. Improved coresets for kernel density estimates. In *Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2718–2727, 2018.
- [8] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 20, 2007.
- [9] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with Rotary Position Embedding. *Neurocomputing*, 568:127063, 2024.
- [10] 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 (NeurIPS)*, volume 30, 2017.
- [11] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations (ICLR)*, 2024.
- [12] Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. H₂O: Heavy-Hitter Oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023.