SLAY:
Spherical Linearized Attention with Yat-Kernel

Jose Miguel Luna [1] **Taha Bouhsine** [2] **Krzysztof Choromanski** [3]

Abstract

The Yat-product operator (q, k) yields geometry-aware interactions but leads to quadratic $O(L^2)$ attention due to a non-factorizable denominator. We introduce *SLAY* (*Spherical Linearized Attention with YAT-Kernels*), which constrains queries and keys to the unit sphere so the kernel depends only on angular alignment. Using Bernstein's theorem, we express the spherical YAT kernel as a nonnegative mixture of polynomial–exponential product kernels and derive a strictly positive random-feature approximation with linear-time $O(L)$ attention. We prove positive definiteness and boundedness on the sphere and show the estimator yields nonnegative scores with a positive denominator under mild conditions. Empirically, SLAY matches full spherical YAT attention and compares favorably to linearized baselines on language modeling and image classification while retaining linear scaling in L.

1. Introduction

Transformer models owe much of their success to attention, which enables dynamic, content-dependent interactions between tokens. In standard Transformers, attention is implemented via softmax applied to pairwise query–key similarities. While expressive, this requires constructing an explicit $L \times L$ attention matrix for a sequence of length L, resulting in quadratic time and memory complexity. This cost rapidly becomes prohibitive as context lengths grow, fundamentally limiting long-context modeling.

[1]Columbia University [2]Azetta.ai [3]Google DeepMind. Correspondence to: Taha Bouhsine <taha@azetta.ai>.

Proceedings of the 43rd International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026.

To overcome this bottleneck, *linear attention* mechanisms reinterpret attention as kernel evaluation and exploit feature maps to reorder computation. Early examples include kernelized Transformers (11) and random-feature approximations of the softmax kernel (1). These works show that positive random features can yield stable and scalable attention for specific kernels.

Our approach linearizes the spherical -product kernel directly rather than applying a softmax to it, and uses positivity-preserving approximations to ensure stable kernel normalization.

Despite these advances, softmax attention remains tied to a specific similarity: the exponential of an inner product. This choice conflates alignment and magnitude, and its unbounded growth requires careful normalization and stabilization. These limitations motivate alternative attention kernels that are geometrically grounded, self-regularizing, and compatible with efficient computation, as explored in activation-free architectures (2).

Neural Matter Networks (NMNs) introduced the -*Product* (also referred to as the Yat-product) (2), a kernel operator inspired by inverse-square interactions in physics:

$$(q, k) = \frac{(q^\top k)^2}{\|q - k\|^2 + \epsilon} \tag{1}$$

where $\epsilon > 0$ ensures numerical stability. Unlike standard dot-product similarity, the -Product explicitly couples two geometric quantities: *alignment* and *proximity*. The squared inner product in the numerator yields an even (sign-symmetric) alignment score, while the inverse-distance denominator penalizes interactions between distant vectors. This ratio yields a self-regularizing response that can suppress irrelevant interactions without requiring explicit activation functions or normalization layers.

From a theoretical perspective, the -Product can be viewed as a kernel-like similarity operator. NMNs constructed as linear combinations of -Product units are universal approximators on compact domains, despite

being entirely activation-free (2). In this work, we focus on a unit-norm (spherical) setting where the resulting isotropic kernel admits a clean positive-definiteness proof and stable, bounded responses.

In the context of attention, the -Product offers a geometry-aware alternative to softmax similarity. It favors tokens that are both aligned and close in representation space. However, the same geometric coupling introduces a computational obstacle: the denominator $\|q-k\|^2 = \|q\|^2 + \|k\|^2 - 2q^\top k$ entangles query and key terms and prevents the factorization required for efficient linear attention. As a result, naive -Product attention still requires explicit pairwise interactions and reverts to quadratic complexity. This mirrors the general limitation for non-factorizable kernels that motivates linearization techniques such as those used in Performer (1).

This paper addresses this limitation for -Attention. We show that by constraining queries and keys to lie on the unit sphere and reformulating the resulting kernel using Bernstein's theorem, the -Product admits a nonnegative mixture representation that can be approximated by strictly positive random features. This yields a linear-time attention mechanism that preserves the core geometric and self-regulating properties of the -Product while enabling scalable long-context Transformers.

Contributions. We introduce SLAY (**Spherical YAT-Attention**), a geometry-aware attention mechanism, with the following characteristics:

- Enforces unit-norm constraints on queries and keys, decoupling alignment and distance.
- Linearizes the spherical -product via an integral representation based on Bernstein's theorem.
- Approximates the resulting kernel using strictly positive Tensor Product Random Features.
- Achieves linear-time $O(L)$ attention while preserving key theoretical properties of NMNs.

2. Methodology

Our goal is to compute -attention with the spherical -product kernel from Eq. (**??**) without forming the $L\times L$ matrix of pairwise interactions. We first normalize queries/keys to the unit sphere so the kernel depends only on $x = \hat{q}^\top \hat{k} \in [-1,1]$. We then linearize the nonfactorizable term $1/(C-2x)$ via a Laplace integral (Bernstein's theorem), discretize the resulting nonnegative mixture using Gauss–Laguerre quadrature, and approximate each product kernel with strictly positive random features. Finally, we apply the standard kernel-attention reordering using the resulting feature map $\widetilde{\Psi}(\cdot)$.

Throughout this work, we approximate the spherical -product (YAT) kernel itself; normalization is performed via kernel sums as in linear attention, not via a softmax nonlinearity.

2.1. Spherical Constraint

We assume $d \geq 2$ throughout, where spherical isotropy theory applies. We normalize inputs to the unit sphere:

$$\hat{q} = \frac{q}{\|q\|}, \qquad \hat{k} = \frac{k}{\|k\|}, \qquad \|\hat{q}\| = \|\hat{k}\| = 1.$$

Expanding the denominator of Eq. (**??**) yields:

$$\|\hat{q}-\hat{k}\|^2 + \epsilon = \|\hat{q}\|^2 + \|\hat{k}\|^2 - 2\hat{q}^\top\hat{k} + \epsilon \tag{2}$$

$$= (2+\epsilon) - 2\hat{q}^\top\hat{k}. \tag{3}$$

Let $x = \hat{q}^\top\hat{k} \in [-1,1]$ and $C = 2+\epsilon$. The spherical -product becomes

$$_{\text{sph}}(\hat{q},\hat{k}) = \frac{x^2}{C-2x}. \tag{4}$$

Thus, the kernel depends only on angular alignment.

Geometric intuition. On the unit sphere $\mathbb{S}^{d-1}$, the squared *chordal* distance is

$$d_{\mathbb{S}^{d-1}}(\hat{q},\hat{k})^2 = 2(1-\hat{q}^\top\hat{k}),$$

so the spherical -product can be written as a distance-regularized alignment score:

$$_{\text{sph}}(\hat{q},\hat{k}) = \frac{\langle\hat{q},\hat{k}\rangle^2}{d_{\mathbb{S}^{d-1}}(\hat{q},\hat{k})^2 + \epsilon}. \tag{5}$$

Additional discussion (including invariances and the connection to isotropic spherical kernels (3)) is deferred to Appendix **??**.

2.2. Integral Linearization

The function $g(y) = 1/y$ is completely monotone on $(0,\infty)$, which by Bernstein's theorem implies the Laplace representation $1/y = \int_0^\infty e^{-sy}\,ds$ (6; 7). To apply this identity to our kernel, we substitute $y = C-2x$ and verify that $y > 0$ for all $x \in [-1,1]$.

Since $x \leq 1$ and $C = 2+\epsilon$, we have $y = C-2x \geq \epsilon > 0$, hence Bernstein's representation applies on the full domain (see Lemma **??** in Appendix **??**).

Using Bernstein's theorem for completely monotone functions, a standard tool in kernel analysis previously used to construct positive random feature approximations for exponential kernels (1),

$$\frac{1}{y} = \int_0^\infty e^{-sy}\, ds, \tag{6}$$

we obtain

$$_{\text{sph}}(\hat{q}, \hat{k}) = x^2 \int_0^\infty e^{-s(C-2x)} ds \tag{7}$$

$$= \int_0^\infty e^{-sC} \left[x^2 e^{2sx} \right] ds. \tag{8}$$

This expresses the spherical -product as a positively weighted mixture of product kernels: namely, a degree-2 polynomial factor $(\hat{q}^\top \hat{k})^2$ multiplied by an exponential dot-product kernel $e^{2s\,\hat{q}^\top \hat{k}}$.

Importantly, the factor x^2 cannot be absorbed into a *nonnegative* Laplace weight over plain exponentials without introducing signed correction terms (see Appendix **??**).

This representation is analogous to the Laplace-transform-based decompositions used to derive unbiased positive random feature estimators for softmax and Gaussian kernels (1).

2.3. Kernel Approximation and Positive Features

2.3.1. Quadrature (Gauss–Laguerre)

We approximate the integral in Eq. (**??**) using R-point Gauss–Laguerre quadrature. We apply Gauss–Laguerre quadrature after the change of variables $t = Cs$, so that $\int_0^\infty e^{-Cs} h(s)\, ds = \frac{1}{C} \int_0^\infty e^{-t} h(t/C)\, dt$:

$$\int_0^\infty e^{-sC} h(s)\, ds \;\approx\; \sum_{r=1}^{R} w_r\, h(s_r),$$

where $\{t_r, \alpha_r\}_{r=1}^R$ are the standard Gauss–Laguerre nodes and weights for $\int_0^\infty e^{-t} f(t)\, dt$ and

$$s_r = \frac{t_r}{C}, \qquad w_r = \frac{\alpha_r}{C}.$$

Thus, the w_r already incorporate the $1/C$ factor induced by $t = Cs$. Since $|x| \le 1$, the integrand $h(s) = x^2 e^{2sx}$ is entire and uniformly bounded by e^{2s}, ensuring uniform exponential convergence over $x \in [-1, 1]$. Such bounds follow from classical results on Gauss–Laguerre quadrature for entire functions of exponential type (see Theorem 3.6.24 in (4; 12)); for a shapes/implementation summary, see Appendix **??**.

2.3.2. Random features (polynomial and exponential)

Polynomial component and approximations. For the exact polynomial kernel $(\hat{q}^\top \hat{k})^2$, the explicit feature map $\phi_{\text{poly}}(u) = \text{vec}(uu^\top) \in \mathbb{R}^{d^2}$ yields exact reconstruction:

$$\langle \phi_{\text{poly}}(\hat{q}), \phi_{\text{poly}}(\hat{k}) \rangle = (\hat{q}^\top \hat{k})^2.$$

In practice, we reduce dimensionality via low-dimensional approximations. We consider TensorSketch (5) as well as three common approximations to the degree-2 polynomial kernel $k_{\text{poly}}(x, y) = (x^\top y)^2$, described formally in Appendix **??**: (i) Random Maclaurin (RM) features (14), (ii) Nystr"om features using anchor points (15), and (iii) anchor features using squared inner products to fixed anchors.

Anchor features. Let anchors $\{a_i\}_{i=1}^P \subset \mathbb{R}^d$ be fixed reference vectors. Anchor features define

$$\phi_{\text{anc}}(x) = \frac{1}{\sqrt{P}} \left[(x^\top a_i)^2 \right]_{i=1}^P,$$

yielding a simple low-rank approximation whose induced inner products are nonnegative.

Unlike Nystr"om approximations, anchor features do not require inversion/whitening of the anchor Gram matrix and therefore preserve non-negativity of approximate kernel evaluations.

Default choice. Unless stated otherwise, we use *anchor features* as the default polynomial approximation because they (i) preserve non-negativity of the polynomial component (supporting the denominator-positivity guarantees), (ii) are empirically stable at small feature budgets, and (iii) are computationally simple ($O(dP)$ per token). The multi-scale sweep in Table **??** (Appendix **??**) supports this choice.

Table **??** summarizes the trade-offs between polynomial approximations, highlighting positivity preservation as a key distinction.

For the theoretical non-negativity and denominator-positivity guarantees stated later, we require a polynomial component whose induced score estimates are nonnegative (e.g., the exact map, or anchor features). Signed polynomial approximations (TensorSketch, Random Maclaurin) and Nystr"om features can yield negative approximate inner products and are therefore treated as accuracy/efficiency baselines rather than positivity-guaranteeing estimators.

Why the polynomial factor is needed for fidelity. The x^2 factor in Eq. (**??**) is part of the tar-

Table 1. Polynomial kernel approximation options for $(x^\top y)^2$. Here D_p denotes the polynomial-feature dimension, and P denotes the number of anchors (when applicable). *Feature cost* is the asymptotic cost of computing the polynomial features for one vector and excludes quadrature/PRF computation, tensor-product fusion/sketching, and the linear-attention contractions; these additional costs drive end-to-end latency in Section **??**.

Method	Dim.	Feature cost	Unbiased?	$\langle\phi(x),\phi(y)\rangle \geq 0$?
Exact vec$(uu^\top)$	d^2	$O(d^2)$	Yes	Yes
TensorSketch (5)	D_p	$\approx O(d + D_p \log D_p)$	Approx.	No (not guaranteed)
Random Maclaurin (14)	D_p	$O(d\,D_p)$	Yes	No (not guaranteed)
Nyström (15)	P	$O(d\,P)$	Approx.	No (not guaranteed)
Anchor features (low-rank) (9)	P	$O(d\,P)$	No	Yes

get kernel; removing it yields a different kernel (see Appendix **??**).

Retaining the polynomial factor is essential both for kernel fidelity and for positivity guarantees of the resulting attention scores.

Positive Random Features. For the exponential term $e^{2s\,\hat{q}^\top \hat{k}}$, we use Positive Random Features applied to the *original normalized vectors*:

$$\phi_{\mathrm{PRF}}(u;s) = \frac{1}{\sqrt{D}}\left[\exp\Big(\sqrt{2s}\,\omega_i^\top u - s\Big)\right]_{i=1}^{D}, \qquad (9)$$

where $\omega_i \sim \mathcal{N}(0, I_d)$ are drawn independently. This construction satisfies

$$\mathbb{E}\Big[\langle \phi_{\mathrm{PRF}}(\hat{q};s), \phi_{\mathrm{PRF}}(\hat{k};s)\rangle\Big] = e^{2s\,\hat{q}^\top \hat{k}}$$

for unit-norm inputs (a standard positive random feature identity for exponential dot-product kernels; see (1) and Proposition **??** in Appendix **??**).

2.3.3. Fusing polynomial and PRF features

Fusion. Since the polynomial and PRF feature maps are applied to vectors (not scalars), we obtain for each scale r:

$$\widetilde{\Psi}_r(u) = \sqrt{w_r}\,\mathcal{S}\Big(\phi_{\mathrm{poly}}(u) \otimes \phi_{\mathrm{PRF}}(u;s_r)\Big), \qquad (10)$$

where $\mathcal{S} : \mathbb{R}^{D_p D_r} \to \mathbb{R}^{D_t}$ is a (randomized) sketching operator that approximates the tensor-product feature map without explicitly materializing the D_pD_r-dimensional Kronecker vector. We then define $\widetilde{\Psi}(u)$ as the concatenation over $r = 1, \ldots, R$.

Conceptually, the target kernel at each quadrature node is the product kernel $(\hat{q}^\top \hat{k})^2\, e^{2s_r \hat{q}^\top \hat{k}}$, whose RKHS is the tensor product $\mathcal{H}_{\mathrm{poly}} \otimes \mathcal{H}_{\exp,s_r}$. The sketching operator $\mathcal{S}$ provides a computationally efficient approximation of this tensor-product feature map.

Remark 1 (Feature Map Target)**.** *$\widetilde{\Psi}$ targets the* integrand $k_s(x) = x^2 e^{2sx}$*; consequently,*

$$\mathbb{E}[\langle \widetilde{\Psi}(\hat{q}), \widetilde{\Psi}(\hat{k})\rangle] \approx \sum_{r=1}^{R} w_r\,(\hat{q}^\top \hat{k})^2 e^{2s_r \hat{q}^\top \hat{k}},$$

which is a quadrature approximation to Eq. (**??**)*.*

Remark 2 (Bias Decomposition)**.** *$\langle \widetilde{\Psi}(\hat{q}), \widetilde{\Psi}(\hat{k})\rangle$ is unbiased for the discretized (quadrature) kernel, but biased for the true kernel unless $R \to \infty$.*

2.4. Linear-Time Attention Computation

Given $Q, K, V \in \mathbb{R}^{L\times d}$, we compute normalized inputs, apply Ψ, and use the standard linear-attention reordering:

$$\hat{Y} = \frac{\widetilde{\Psi}(Q)\big(\widetilde{\Psi}(K)^\top V\big)}{\widetilde{\Psi}(Q)\big(\widetilde{\Psi}(K)^\top \mathbf{1}\big)}.$$

Note that this normalization is not a softmax; it corresponds to kernel normalization and preserves linear-time computation. Here the division is applied rowwise (broadcast across the value dimension); in practice we add a small stabilizer $\delta > 0$ to the denominator for numerical stability (13). Concretely, if $\Psi(Q), \Psi(K) \in \mathbb{R}^{L\times m}$ (feature dimension m) and $V \in \mathbb{R}^{L\times d_v}$, then $\Psi(K)^\top V \in \mathbb{R}^{m\times d_v}$, $\Psi(K)^\top \mathbf{1} \in \mathbb{R}^{m\times 1}$, and the denominator is an $L \times 1$ vector broadcast over d_v (see Appendix **??**).

Complexity. Let m denote the final feature dimension (after concatenating across R quadrature nodes), which in our construction scales as $m = O(RD_t)$. The linear-attention contractions cost $O(L\,m\,d_v)$ time and $O(L\,m)$ space (the $L \times L$ attention matrix is never formed). Feature construction depends on the polynomial approximation: exact degree-2 features cost $O(L\,d^2)$, anchor features cost $O(L\,dP)$, Random

Algorithm 1 Spherical -Attention Forward Pass

Require: $Q, K, V \in \mathbb{R}^{L \times d}$
Normalize Q, K to unit norm
Compute polynomial features $\phi_{\text{poly}}(Q), \phi_{\text{poly}}(K)$ (e.g., anchor features by default)
for $r = 1$ to R **do**
  Compute PRF features $\phi_{\text{PRF}}(\cdot; s_r)$
  Fuse features via sketched tensor product $\widetilde{\Psi}_r$
end for
Concatenate features $\widetilde{\Psi}(Q), \widetilde{\Psi}(K)$
Compute numerator and denominator
return $\hat{Y}$

Maclaurin costs $O(L\, dD_p)$, and TensorSketch costs approximately $O(L\,(d + D_p \log D_p))$ per layer. The PRF exponential features contribute $O(L\, R\, dD)$ time. Appendix **??** collects practical knob choices and shape details. See Appendix **??** for explicit tensor-product scaling (without sketching) and causal-vs.-non-causal implementation notes.

3. Experiments

We evaluate Spherical -Attention along three axes. First, we run one-epoch end-to-end training for base encoder (BERT-style) and decoder (GPT-style) models to verify optimization stability under a fixed training recipe. Second, we benchmark a single attention layer to measure latency and memory scaling with sequence length. Third, we measure full-model inference behavior and the maximum context length attainable under a fixed GPU memory budget.

Unless stated otherwise, the softmax and Spherical variants share the same architecture and training/inference settings; we change only the attention operator and its feature configuration.

3.1. Ablation: Polynomial Approximation Choices

We isolate the polynomial factor $(\hat{q}^\top \hat{k})^2$ and compare several approximations (anchor, Nyström, TensorSketch, Random Maclaurin). We report (i) kernel-normalized attention output error relative to exact spherical -attention and (ii) forward-pass latency, under matched feature budgets; protocol details and the multi-scale sweep are in Appendix **??** and Table **??** (Appendix **??**).

For reference, we also include Laplace-only and a Hadamard-fusion variant. These are not polynomial approximations of $(\hat{q}^\top \hat{k})^2$: they change the estimator by removing or altering the polynomial factor. Under feature-budget matching, they can therefore require substantially larger PRF feature dimension to reach comparable overall feature counts, which can dominate runtime.

Table 2. Polynomial-approximation ablation (large-scale snapshot; see the "Large" block in Table **??**). Lower Rel. ℓ_2 is better; latency is forward-pass time under the same feature-budget matching across methods (which can increase PRF dimensionality for Laplace-only/Hadamard). Very large errors indicate severe kernel mismatch/instability. Full multi-scale sweep is in Table **??** (Appendix **??**).

Method	Rel. $\ell_2 \downarrow$	Latency (ms)$\downarrow$
Exact (Spherical)	0.0000	5.02
Laplace-only	0.4850	1905.80
Anchor	0.4939	489.42
Hadamard (shared ω)	0.6793	1932.07
Nyström	28.1970	569.64
TensorSketch	4.62×10^5	547.76
Random Maclaurin	1.77×10^6	551.43

At the large scale (Table **??**), anchor and Laplace-only achieve the lowest errors, but anchor is markedly faster (489 ms vs. 1906 ms) and is our default choice in the remaining experiments. Nyström is substantially less accurate at these budgets. Signed polynomial approximations (TensorSketch and Random Maclaurin) can yield negative approximate inner products, leading to denominator cancellation and severe instability; we include them as efficiency baselines rather than positivity-guaranteeing estimators.

3.2. Computational Costs

We compare the computational efficiency of **SLAY**, a linearized estimator of spherical YAT attention, against both quadratic and linear attention mechanisms. We report latency, peak memory usage, and throughput as functions of sequence length, focusing on regimes relevant to long-context modeling.

Benchmark setup. All attention mechanisms are benchmarked in isolation using a causal attention kernel with identical architectural settings (embedding dimension 256, 8 heads, batch size 1). Experiments are conducted on a single NVIDIA A100-SXM4 GPU (80 GB). For each sequence length, we report mean latency over multiple timed runs after warm-up, peak GPU memory allocation, and effective throughput measured in tokens per second. Sequence lengths range from 128 tokens up to 131K tokens, or until out-of-memory (OOM) failure.

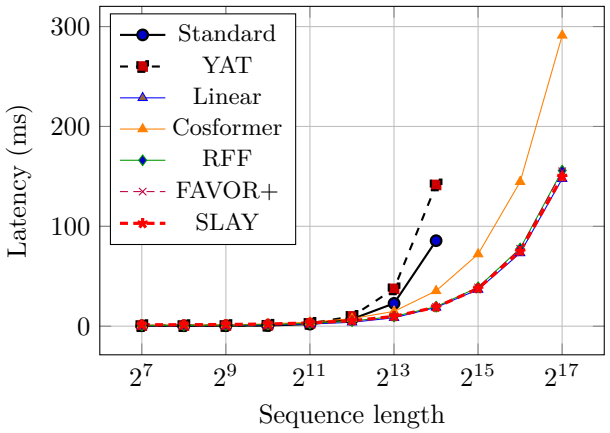

Figure 1. Latency versus sequence length. Quadratic attention mechanisms scale poorly and fail beyond 16K tokens, while all linear methods—including SLAY—exhibit linear scaling up to 128K tokens.

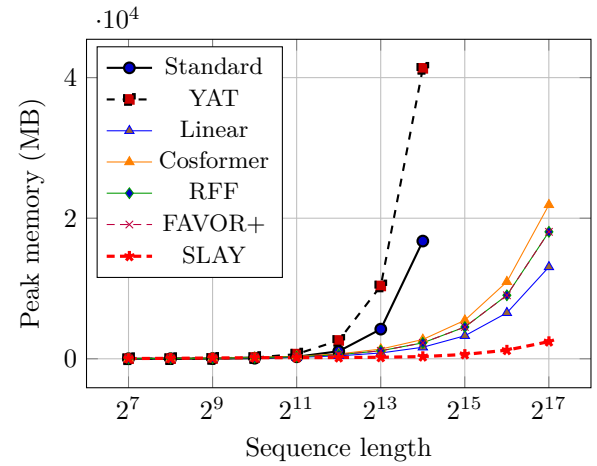

Figure 2. Peak GPU memory usage versus sequence length. Exact quadratic methods exhibit quadratic memory growth and fail beyond 16K tokens, while SLAY maintains a low and strictly linear memory footprint.

Results overview. Figures **??**–**??** summarize the scaling behavior. Quadratic attention mechanisms (standard softmax and exact YAT) exhibit rapidly increasing latency and memory usage, failing beyond 16K tokens. In contrast, linear attention methods scale approximately linearly and remain stable up to 128K tokens.

SLAY closely follows the scaling behavior of other linear attention mechanisms while substantially reducing memory usage relative to exact spherical YAT attention. At long sequence lengths, SLAY sustains high throughput comparable to other linear baselines, demonstrating that its added geometric structure does not compromise scalability.

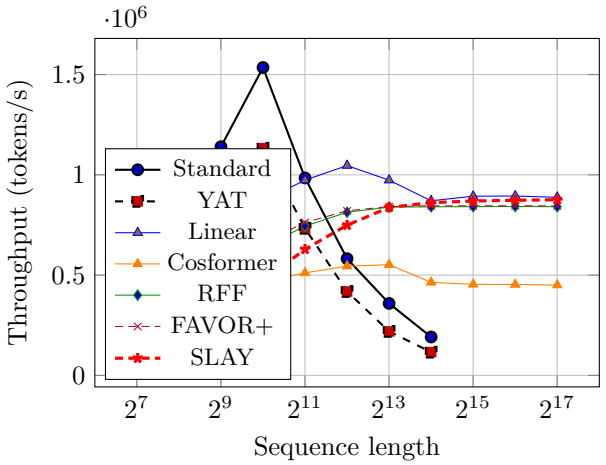

Figure 3. Throughput versus sequence length. Linear attention methods—including SLAY—maintain high throughput at long context lengths, while quadratic methods rapidly degrade and become infeasible.

3.3. Attention Approximation Errors

3.4. Synthetic Tasks Experimentation

Table 3. Synthetic task performance (Basic). Accuracy (mean ± std over 3 seeds).

Method	Copy	Sort	Reverse
Standard	1.00±0.00	0.28±0.02	0.51±0.00
Spherical–YAT	1.00±0.00	0.24±0.01	0.33±0.02
Performer	1.00±0.00	0.27±0.02	0.09±0.01
Linear	1.00±0.00	0.26±0.02	0.05±0.01
SLAY	1.00±0.00	0.29±0.01	0.42±0.04

Table 4. Synthetic task performance (Arithmetic). Accuracy (mean ± std over 3 seeds).

Method	Counting	Parity	Addition	Modular
Standard	0.72±0.01	0.49±0.03	0.78±0.03	0.15±0.03
Spherical–YAT	0.78±0.04	0.49±0.03	0.68±0.16	0.16±0.01
Performer	0.81±0.06	0.49±0.03	0.84±0.02	0.15±0.02
Linear	0.83±0.05	0.49±0.03	0.91±0.04	0.15±0.03
SLAY	0.74±0.13	0.49±0.03	0.86±0.05	0.20±0.03

3.5. Extreme Classification Test

3.6. SLAYformer: Testing SLAY on Transformer Architectures

Comparing performance of 2.5B parameter models with different attention mechanisms.

Comparing TEXT next token prediction and image classification (ViT)

Detail inference scaling results.

Table 5. Synthetic task performance (Long-Range). Accuracy (mean ± std over 3 seeds).

Method	Long Copy	Distant Match	Multihop
Standard	1.00±0.00	1.00±0.00	0.04±0.01
Spherical–YAT	1.00±0.00	1.00±0.00	0.04±0.01
Performer	1.00±0.00	1.00±0.00	0.03±0.02
Linear	1.00±0.00	0.99±0.02	0.03±0.00
SLAY	1.00±0.00	1.00±0.00	0.04±0.01

Table 6. Synthetic task performance (Memory). Accuracy (mean ± std over 3 seeds).

Method	Retrieval	Kv Recall	First Token	Selective Copy
Standard	1.00±0.00	0.02±0.02	1.00±0.00	0.88±0.00
Spherical–YAT	1.00±0.00	0.02±0.03	1.00±0.00	0.88±0.00
Performer	1.00±0.00	0.03±0.00	1.00±0.00	0.88±0.00
Linear	1.00±0.00	0.02±0.02	0.97±0.02	0.88±0.00
SLAY	1.00±0.00	0.02±0.01	1.00±0.00	0.88±0.00

Table 7. Synthetic task performance (Patterns). Accuracy (mean ± std over 3 seeds).

Method	Bigram	Majority	Histogram
Standard	–	0.78±0.07	0.87±0.00
Spherical–YAT	–	0.75±0.06	0.87±0.00
Performer	–	0.82±0.02	0.87±0.00
Linear	–	0.82±0.06	0.87±0.00
SLAY	–	0.84±0.03	0.87±0.00

Table 8. Synthetic task performance (Reasoning). Accuracy (mean ± std over 3 seeds).

Method	Stack	Induction	Pattern
Standard	0.75±0.01	0.02±0.02	0.91±0.00
Spherical–YAT	0.75±0.01	0.02±0.02	0.91±0.00
Performer	0.76±0.01	0.02±0.01	0.91±0.00
Linear	0.75±0.01	0.01±0.01	0.91±0.00
SLAY	0.76±0.01	0.03±0.01	0.91±0.00

Table 9. Synthetic task performance (Robustness). Accuracy (mean ± std over 3 seeds).

Method	Noisy Copy	Compression
Standard	1.00±0.00	0.59±0.00
Spherical–YAT	1.00±0.00	0.59±0.00
Performer	1.00±0.00	0.59±0.01
Linear	1.00±0.00	0.59±0.01
SLAY	1.00±0.00	0.59±0.01

4. Discussion and Limitations

This work provides a directional but concrete demonstration that the -product attention mechanism from Neural Matter Networks can be scaled to long contexts without sacrificing its core theoretical properties.

Scale of experiments. Our experiments are intentionally conservative, focusing on single-epoch training and base-sized models. Larger-scale experiments—including longer training runs, larger models, and multi-node setups—are necessary to fully assess downstream task performance and convergence behavior at scale.

Constant-factor overhead. Spherical -Attention introduces higher constant latency than softmax attention at short sequence lengths due to random-feature computation. As a result, it is not intended to replace softmax universally, but rather to enable regimes where quadratic attention is infeasible.

Architectural generality. While we focus on Transformer-based language models, the underlying linearization applies to any architecture using -product interactions. Promising future directions include applying -Attention to:

- Reinforcement learning models with long-horizon credit assignment,
- Memory-augmented agents and world models,
- Physics-inspired neural architectures beyond Transformers.

Approximation error. The TP-PRF approximation introduces bias controlled by feature dimension and quadrature resolution. Understanding how this bias interacts with optimization dynamics remains an open research question.

5. Conclusion

We introduced the *Spherical -Attention*, a linear-time attention mechanism that makes the -product operator from Neural Matter Networks practical for long-context sequence modeling. By enforcing unit-norm constraints, applying Bernstein's theorem, and approximating the resulting kernel with strictly positive Tensor Product Random Features, we obtain a factorizable and bounded attention kernel compatible with FAVOR$^+$-style computation.

Theoretical analysis shows that the proposed approximation preserves the self-regulation and superposition

properties of the original -product. Empirically, we demonstrate stable end-to-end training, linear time and memory scaling, and the ability to process sequences 30X longer than standard attention on a single 80 GB GPU.

Taken together, these results suggest that linearized -Attention offers a viable path toward scalable, geometry-aware attention mechanisms, bridging Neural Matter Networks and practical long-context Transformers.

More broadly, this work also demonstrates that attention mechanisms need not be restricted to softmax-like similarities to admit linear-time computation. By combining spherical geometry with general kernel linearization tools, Spherical Yat-Attention opens a path toward scalable attention mechanisms grounded in alternative interaction principles.

References

[1] K. Choromanski et al. Rethinking attention with performers. In *ICLR*, 2021.

[2] Taha Bouhsine No More DeLuLu: A Kernel-Based Activation-Free Neural Networks.

[3] I. J. Schoenberg. Positive definite functions on spheres. *Duke Mathematical Journal*, 9(1):96–108, 1942.

[4] P. J. Davis and P. Rabinowitz. *Methods of Numerical Integration*. Academic Press, 2nd edition, 1984.

[5] N. Pham and R. Pagh. Fast and scalable polynomial kernels via explicit feature maps. In *KDD*, 2013.

[6] D. V. Widder. *The Laplace Transform*. Princeton University Press, 1941.

[7] R. L. Schilling, R. Song, and Z. Vondraček. *Bernstein Functions: Theory and Applications*. De Gruyter, 2nd edition, 2012.

[8] N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3):337–404, 1950.

[9] B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, 2002.

[10] A. Berlinet and C. Thomas-Agnan. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Kluwer Academic Publishers, 2004.

[11] S. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *ICML*, 2020.

[12] W. Gautschi. *Orthogonal Polynomials: Computation and Approximation*. Oxford University Press, 2004.

[13] N. J. Higham. *Accuracy and Stability of Numerical Algorithms*. SIAM, 2nd edition, 2002.

[14] A. Kar and H. Karnick. Random feature maps for dot product kernels. In *AISTATS*, 2012.

[15] C. K. I. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. In *NIPS*, 2001.

Appendix

A. Background Results

This section collects standard background facts used in the main text.

Lemma 1 (Bernstein Representation Applicability). *For $\epsilon > 0$ and $C = 2 + \epsilon$, the variable $y = C - 2x$ satisfies $y \geq \epsilon > 0$ for all $x \in [-1, 1]$. Hence Bernstein's representation $1/(C - 2x) = \int_0^\infty e^{-s(C-2x)}\, ds$ applies throughout the domain.*

Proof. See Appendix **??**. □

B. Geometric Interpretation and Invariances

This appendix expands on the geometric view of the spherical -product discussed in Section **??**. On $\mathbb{S}^{d-1}$, the squared chordal distance satisfies $d_{\mathbb{S}^{d-1}}(\hat{q}, \hat{k})^2 = 2(1 - \hat{q}^\top \hat{k})$, so the spherical kernel can be interpreted as an ϵ-regularized chordal-distance interaction.

Proposition 1 (Geometric Origin). *The spherical -product is an ϵ-regularized chordal-distance interaction on $\mathbb{S}^{d-1}$:*

$$_{\text{sph}}(\hat{q}, \hat{k}) = \frac{\langle \hat{q}, \hat{k} \rangle^2}{d_{\mathbb{S}^{d-1}}(\hat{q}, \hat{k})^2 + \epsilon},$$

where the numerator captures directional alignment and the denominator enforces locality via chordal proximity.

Since the kernel depends only on $\hat{q}^\top \hat{k}$, it belongs to the class of isotropic kernels on the sphere. All spherical positive-definiteness claims in the main text assume $d \geq 2$, where Schoenberg's characterization applies (3).

Remark 3 (Geometric Invariances). *The spherical -product is invariant under (i) rotations: $_{\text{sph}}(R\hat{q}, R\hat{k}) = _{\text{sph}}(\hat{q}, \hat{k})$ for all $R \in SO(d)$, and (ii) uniform scaling prior to normalization. Note that while $(\hat{q}^\top \hat{k})^2$ is even under sign flips, the full kernel $_{\text{sph}}(\hat{q}, \hat{k}) = \frac{(\hat{q}^\top \hat{k})^2}{(2+\epsilon)-2\hat{q}^\top \hat{k}}$ is not invariant under $\hat{q} \mapsto -\hat{q}$ in general.*

Proposition 2 (PRF Unbiasedness). *For $\hat{q}, \hat{k} \in \mathbb{S}^{d-1}$ and $\{\omega_i\}_{i=1}^D \overset{\text{iid}}{\sim} \mathcal{N}(0, I_d)$:*

$$\mathbb{E}\left[\left\langle \phi_{PRF}(\hat{q}; s), \phi_{PRF}(\hat{k}; s) \right\rangle\right] = e^{2s\hat{q}^\top \hat{k}}.$$

Proof. See Appendix **??**. □

Lemma 2 (Positive Mixture Closure). *If $\{k_s\}_{s \geq 0}$ is a family of positive-definite kernels on $\mathcal{X}$ and $w(s) \geq 0$ is a nonnegative measure, then $k(x, y) = \int_0^\infty w(s) k_s(x, y)\, ds$ is PD on $\mathcal{X}$ (a standard closure property of PD kernels; see, e.g., (8; 9)).*

Theorem 1 (Tensor Kernel Decomposition). *Let k_1, k_2 be positive-definite kernels on $\mathcal{X}$ with RKHSs $\mathcal{H}_1, \mathcal{H}_2$. Then the product kernel $k(x, y) = k_1(x, y) \cdot k_2(x, y)$ is positive definite with RKHS $\mathcal{H} = \mathcal{H}_1 \otimes \mathcal{H}_2$ (see, e.g., (8; 10)).*

C. Preliminaries: Polynomial Kernel Approximations

We summarize three approximations of the degree-2 polynomial kernel $k_{\text{poly}}(x, y) = (x^\top y)^2$ used in our implementation and ablations. Let $x, y \in \mathbb{R}^d$.

Random Maclaurin (RM). Draw Rademacher vectors $r_i, s_i \in \{\pm 1\}^d$ and define

$$\phi_{\text{RM}}(x) = \frac{1}{\sqrt{P}}\big[(r_i^\top x)(s_i^\top x)\big]_{i=1}^{P}.$$

Then $\mathbb{E}[\langle \phi_{\text{RM}}(x), \phi_{\text{RM}}(y)\rangle] = (x^\top y)^2$ (14). RM is unbiased but can have high variance for small P.

Nystr"om features. Let anchors $A = \{a_1, \ldots, a_P\} \subset \mathbb{R}^d$ and define $K_{AA} \in \mathbb{R}^{P\times P}$ with $(K_{AA})_{ij} = (a_i^\top a_j)^2$. Given $K_{AA} + \lambda I$, define

$$\phi_{\text{Nys}}(x) = (K_{xA})(K_{AA} + \lambda I)^{-1/2}, \quad K_{xA} = [(x^\top a_i)^2]_{i=1}^{P}.$$

This yields a low-rank approximation whose quality depends on anchor coverage and conditioning (15).

Anchor (low-rank) features. Using the same anchors A, define

$$\phi_{\text{Anc}}(x) = \frac{1}{\sqrt{P}}\big[(x^\top a_i)^2\big]_{i=1}^{P}.$$

This is a simple low-rank approximation; it is not unbiased in general but is often stable for small P.

Comparison. RM is unbiased but variance-dominated at small feature budgets; Nystr"om reduces variance if anchors are well-conditioned; anchor features are computationally simplest and empirically most stable at small P.

D. Ablation: Polynomial Approximation Sweep

This appendix reports the multi-scale kernel-fidelity sweep referenced in Section **??**. All variants tie the QKV and output projection weights and compare outputs against exact kernel-normalized spherical -attention.

E. Integral Representation of the Spherical -Product

In this appendix, we provide a detailed derivation of the integral representation used to linearize the spherical -product kernel.

Recall that under the unit-norm constraint $\hat{q}, \hat{k} \in \mathbb{S}^{d-1}$, the -product reduces to

$$_{\text{sph}}(\hat{q}, \hat{k}) = \frac{x^2}{C - 2x}, \qquad x = \hat{q}^\top \hat{k} \in [-1, 1], \quad C = 2 + \epsilon. \tag{11}$$

The function $g(y) = 1/y$ is completely monotone on $(0, \infty)$ and therefore admits the Bernstein representation

$$\frac{1}{y} = \int_0^\infty e^{-sy}\, ds. \tag{12}$$

Applying this identity with $y = C - 2x$, we obtain

$$_{\text{sph}}(\hat{q}, \hat{k}) = x^2 \int_0^\infty e^{-s(C-2x)}\, ds \tag{13}$$

$$= \int_0^\infty e^{-sC}\, x^2 e^{2sx}\, ds. \tag{14}$$

This representation expresses the spherical -product as a positively weighted mixture of a degree-2 polynomial kernel and an exponential kernel in the angular variable x. This decomposition forms the basis for the random-feature approximation introduced in the main text.

Table 10. Multi-scale ablation over feature budgets for polynomial-kernel approximations. We compare attention outputs against *exact kernel-normalized* spherical YAT with tied QKV/out projections. Lower Rel. ℓ_2 is better; latency is forward-pass time.

Scale	Method	T	R	M	P	Rel. $\ell_2 \downarrow$	Latency (ms)$\downarrow$
Small	Exact (Spherical)	128	2	8	8	0.0000	3.12
	Laplace-only					0.5870	2.78
	Anchor					0.6626	3.82
	Hadamard (shared ω)					0.8237	3.34
	Nyström					22.9072	3.41
	TensorSketch					474075.1562	5.17
	Random Maclaurin					2195912.7500	5.59
Medium	Exact (Spherical)	256	2	16	16	0.0000	0.79
	Laplace-only					0.5417	16.10
	Anchor					0.5667	18.54
	Hadamard (shared ω)					0.6609	17.44
	Nyström					61.6529	18.46
	TensorSketch					214115.9844	19.01
	Random Maclaurin					1715766.8750	18.80
Large	Exact (Spherical)	512	2	32	32	0.0000	5.02
	Laplace-only					0.4850	1905.80
	Anchor					0.4939	489.42
	Hadamard (shared ω)					0.6793	1932.07
	Nyström					28.1970	569.64
	TensorSketch					461739.0312	547.76
	Random Maclaurin					1772757.5000	551.43

F. Random Feature Construction and Unbiasedness

In this appendix, we provide additional details on the random-feature construction used to approximate the integrand appearing in the spherical -product representation and justify its unbiasedness.

Recall from Eq. (**??**) that the spherical -product admits the decomposition

$$_{\mathrm{sph}}(\hat{q}, \hat{k}) = \int_0^\infty e^{-sC}\, x^2 e^{2sx}\, ds, \qquad x = \hat{q}^\top \hat{k}.$$

Polynomial component. The term $x^2 = (\hat{q}^\top \hat{k})^2$ corresponds to a homogeneous degree-2 polynomial kernel. This kernel admits an explicit feature map given by

$$\phi_{\mathrm{poly}}(u) = \mathrm{vec}(uu^\top),$$

or an approximate variant implemented via tensor sketching. In both cases, the inner product of feature maps yields an unbiased estimator of the polynomial kernel.

Exponential component. The exponential term e^{2sx} is approximated using strictly positive random features. For random projections ω drawn from a Gaussian or orthogonal distribution, the feature map

$$\phi_{\mathrm{PRF}}(u; s) = \frac{1}{\sqrt{D}} \exp\Big(\sqrt{2s}\, \omega^\top u - s\Big)$$

satisfies

$$\mathbb{E}\big[\langle \phi_{\mathrm{PRF}}(\hat{q}; s), \phi_{\mathrm{PRF}}(\hat{k}; s)\rangle\big] = e^{2s\, \hat{q}^\top \hat{k}}.$$

Tensor product approximation. Since the polynomial component can be computed exactly (or approximated in practice) and the exponential component is estimated with unbiased PRFs, their tensor product

$$\phi_{\mathrm{poly}}(u) \otimes \phi_{\mathrm{PRF}}(u; s)$$

is an unbiased estimator of the product kernel $x^2 e^{2sx}$ by linearity of expectation. Approximating the outer integral using quadrature preserves unbiasedness up to the discretization error introduced by the numerical integration scheme.

On "pure Laplace" forms. If one insists on a nonnegative Laplace mixture of plain exponentials e^{2sx}, then $x^2/(C-2x)$ cannot be represented exactly, because $k(0) = 0$ whereas $\int_0^\infty w(s)\, e^{2s\cdot 0}\, ds = \int_0^\infty w(s)\, ds \geq 0$ for any $w \geq 0$. There is, however, an exact decomposition that removes the explicit x^2 factor at the cost of an affine correction term:

$$\frac{x^2}{C-2x} = \frac{C^2}{4} \int_0^\infty e^{-Cs}\, e^{2sx}\, ds \;-\; \frac{C}{4} \;-\; \frac{x}{2}.$$

This identity follows from $x^2 e^{2sx} = \frac{1}{4}\,\partial_s^2 e^{2sx}$ and two integrations by parts, whose boundary terms yield the affine correction. While this removes the need for polynomial random features, it introduces signed components (through the subtraction of constant and linear kernels) and therefore does not retain the "strictly positive feature map" and denominator-positivity guarantees emphasized in the main construction.

Hadamard (elementwise) fusion variant. Some implementations replace the tensor product with elementwise (Hadamard) fusion,

$$\phi_{\mathrm{had}}(u; s) = \sqrt{w_r}\,\big(\phi_{\mathrm{poly}}(u) \odot \phi_{\mathrm{PRF}}(u; s)\big),$$

which yields a valid positive feature map but targets a *different* kernel than the tensor product. In particular, the expected inner product becomes

$$\mathbb{E}[\langle \phi_{\mathrm{had}}(\hat{q}; s), \phi_{\mathrm{had}}(\hat{k}; s)\rangle] \;\approx\; (\hat{q}^\top \hat{k})^2\, e^{2s\,\hat{q}^\top \hat{k}} \quad \text{only if } \phi_{\mathrm{poly}} \text{ and } \phi_{\mathrm{PRF}} \text{ are aligned feature maps.}$$

With standard independent random features, Hadamard fusion instead corresponds to a product of marginal kernels across matched feature indices, which generally introduces bias relative to the target integrand kernel. The benefit is computational: it avoids the $D_p \times D_r$ tensor expansion and reduces memory, but at the cost of a kernel mismatch. We therefore treat Hadamard fusion as a fast baseline rather than the primary estimator of the spherical -product.

G. Positivity and Stability Guarantees

This appendix provides additional justification for the positivity and numerical stability properties of the proposed linearized -attention mechanism.

Positivity. All components of the *target* spherical kernel are non-negative. Moreover, if the polynomial feature map is computed exactly (or with a positivity-preserving approximation), then the corresponding approximate scores are non-negative:

- The polynomial term $(\hat{q}^\top \hat{k})^2$ is non-negative for all $\hat{q}, \hat{k} \in \mathbb{S}^{d-1}$.
- The exponential term e^{2sx} is strictly positive for all $s \geq 0$ and $x \in [-1, 1]$.
- The quadrature weights w_r are non-negative.

Consequently, under this condition the approximate attention scores produced by the tensor-product random features are non-negative.

Numerical stability. The boundedness of the spherical -product on $\mathbb{S}^{d-1}$ (Proposition **??**) implies that attention scores remain uniformly bounded. Combined with positivity, this prevents the numerical instabilities associated with oscillatory random features and negative attention weights. This behavior mirrors stability properties previously observed in positive random feature–based linear attention mechanisms (1).

H. Experimental and Implementation Details

This appendix summarizes additional experimental and implementation details to facilitate reproducibility.

Table 11. Maximum sequence length for GPT-under 80 GB GPU memory (inference).

Sequence Length	Peak Memory (MB)	Status
4096	11,043	ok
8192	12,225	ok
16384	14,496	ok
32768	20,855	ok
65536	47,626	ok
131072	—	OOM

Random feature configuration. Unless otherwise stated, all experiments use a fixed number of random features per attention head. Quadrature nodes and weights are chosen using standard numerical integration schemes and shared across heads and layers.

Normalization. Queries and keys are explicitly normalized to unit norm prior to feature computation. This normalization is applied per attention head and does not introduce additional learnable parameters.

Hardware and software. All experiments were conducted using PyTorch on NVIDIA A100 GPUs with 80 GB of memory. Attention-only benchmarks use custom linear-attention operators, while full-model experiments rely on standard PyTorch modules augmented with the proposed attention mechanism.

Training configuration. Optimizer settings, learning rates, batch sizes, and training schedules are kept identical across softmax and -based attention variants unless otherwise specified. This ensures that observed differences are attributable to the attention mechanism rather than auxiliary training effects.

Ablation protocol (polynomial approximations). The kernel-fidelity ablation in Section **??** is implemented in `tests/ablation_poly_approx.py`. Running the script with `--sweep` produces a LaTeX table in `tables/poly_ablation_sweep.tex`. All variants tie the QKV and output projection weights and compare outputs against exact kernel-normalized spherical -attention.

Maximum context under fixed memory. Table **??** reports the raw values used to produce Fig. **??**.

Code availability. An open-source implementation of Spherical -Attention, including training scripts and experimental configurations, is available at `https://github.com/jomilu93/Spherical-Yat-Performer.git`.

I. Implementation Notes: Quadrature Scaling and Shapes

Practical knobs and defaults. In our implementation, R controls the quadrature accuracy of the Laplace integral and D controls the Monte Carlo variance of PRF. The polynomial approximation uses either a feature dimension D_p (e.g., Random Maclaurin or TensorSketch) or an anchor count P (anchor features or Nystr"om); by default we use anchor features because they preserve non-negativity of the polynomial component and are stable at small budgets. The tensor-product fusion uses a sketch dimension D_t, trading accuracy for compute/memory. We use small stabilizers ϵ (kernel) and δ (attention denominator) for numerical robustness.

Remark (explicit tensor product). Without sketching, the per-node tensor-product feature dimension is $D_p D$ and the resulting attention cost would scale as $O(L\,R\,D_p D)$ rather than $O(L\,R\,D_t)$. We use sketching to avoid explicitly materializing Kronecker vectors while preserving the same target product-kernel structure up to controlled approximation error.

Causal vs. non-causal. The linearization applies to both causal and non-causal attention. In experiments we use a causal prefix-sum implementation for autoregressive models; for non-causal settings the same features can be used with the standard linear-attention reordering.

J. Mathematical Tools Used (High Level)

Our linearization relies on (i) the Laplace/Bernstein representation of $1/y$ for completely monotone functions (6; 7), (ii) closure properties of positive-definite (PD) kernels under products and nonnegative mixtures (8; 9), (iii) the Gaussian moment generating function to obtain unbiased positive random features for exponential dot-product kernels (1), and (iv) Gauss–Laguerre quadrature to discretize $\int_0^\infty e^{-t} f(t)\, dt$ (4; 12). Numerical stabilization follows standard practice (13).

Gauss–Laguerre scaling. Let $\{t_r, \alpha_r\}_{r=1}^R$ be the Gauss–Laguerre nodes and weights for $\int_0^\infty e^{-t} f(t)\, dt$. With $t = Cs$ and $C = 2 + \epsilon$, we use

$$s_r = \frac{t_r}{C}, \qquad w_r = \frac{\alpha_r}{C},$$

so that $\int_0^\infty e^{-Cs} h(s)\, ds \approx \sum_{r=1}^R w_r h(s_r)$.

Linear-attention shapes. If $\Psi(Q), \Psi(K) \in \mathbb{R}^{L\times m}$ and $V \in \mathbb{R}^{L\times d_v}$, then

$$N = \Psi(Q)\big(\Psi(K)^\top V\big) \in \mathbb{R}^{L\times d_v}, \qquad d = \Psi(Q)\big(\Psi(K)^\top \mathbf{1}\big) \in \mathbb{R}^{L\times 1},$$

and we compute $\hat{Y}_i = N_i/(d_i + \delta)$ with row-wise broadcasting across d_v.

K. Additional Proofs

Proposition 3 (Boundedness on the Unit Sphere)**.** *Let $\hat{q}, \hat{k} \in \mathbb{S}^{d-1}$. Then the spherical -product satisfies*

$$0 \;\le\; {}_{\text{sph}}(\hat{q}, \hat{k}) \;\le\; \frac{1}{\epsilon}.$$

Proposition 4 (Gradient Stability)**.** *There exists a constant C_ϵ such that for all $\hat{q}, \hat{k} \in \mathbb{S}^{d-1}$:*

$$\|\nabla_{\hat{q}} \text{sph}(\hat{q}, \hat{k})\| \;\le\; C_\epsilon.$$

Theorem 2 (Positive Definiteness on $\mathbb{S}^{d-1}$)**.** *For all $d \ge 2$ and $\epsilon > 0$, the spherical -product $k(x) = x^2/(C - 2x)$ with $x \in [-1, 1]$ and $C = 2 + \epsilon$ is a positive-definite kernel on $\mathbb{S}^{d-1}$.*

Proof of Lemma ??.

Proof. Since $x \le 1$ and $C = 2 + \epsilon$, we have $C - 2x \ge C - 2 = \epsilon > 0$. The function $g(y) = 1/y$ is completely monotone on $(0, \infty)$ (all derivatives alternate in sign: $g^{(n)}(y) = (-1)^n n!/y^{n+1}$), so Bernstein's theorem applies and yields $1/(C - 2x) = \int_0^\infty e^{-s(C-2x)}\, ds$. □

Proof of Proposition ??.

Proof.

$$\begin{aligned}
\mathbb{E}\left[\left\langle \phi_{\text{PRF}}(\hat{q}; s), \phi_{\text{PRF}}(\hat{k}; s)\right\rangle\right] &= \frac{1}{D}\sum_{i=1}^D \mathbb{E}\left[\exp\left(\sqrt{2s}\,\omega_i^\top(\hat{q} + \hat{k}) - 2s\right)\right] \\
&= e^{-2s} \cdot \exp\left(s\|\hat{q} + \hat{k}\|^2\right) \\
&= e^{-2s} \cdot e^{s(2 + 2\hat{q}^\top \hat{k})} \\
&= e^{2s\hat{q}^\top \hat{k}}.
\end{aligned}$$

The unit-norm constraint $\|\hat{q}\| = \|\hat{k}\| = 1$ is essential; otherwise, additional norm terms appear in $\|\hat{q} + \hat{k}\|^2$. □

Proof of Proposition ??.

Proof. Let $x = \hat{q}^\top \hat{k} \in [-1, 1]$ and define $f(x) = x^2/(C - 2x)$. Since $C - 2x \geq \epsilon > 0$ on $[-1, 1]$, we have $f(x) \geq 0$. Moreover,

$$f'(x) = \frac{2x(C - x)}{(C - 2x)^2}.$$

On $[-1, 1]$, the maximum of f is attained at $x = 1$, giving $f(1) = 1/\epsilon$. □

Proof of Proposition ??.

Proof. Write $_{\text{sph}}(\hat{q}, \hat{k}) = f(x)$ with $x = \hat{q}^\top \hat{k}$ and $f(x) = x^2/(C - 2x)$. Differentiating gives

$$f'(x) = \frac{2x(C - x)}{(C - 2x)^2}.$$

On $x \in [-1, 1]$ with $C = 2 + \epsilon$, the denominator satisfies $(C - 2x)^2 \geq \epsilon^2$ and the numerator is bounded, hence $|f'(x)| \leq C'_\epsilon$ for some constant depending only on ϵ. By the chain rule, $\nabla_{\hat{q}} \text{sph}(\hat{q}, \hat{k}) = f'(x) \nabla_{\hat{q}} x$ and $\|\nabla_{\hat{q}} x\| = \|\hat{k}\| = 1$ (or its tangent-space projection has norm ≤ 1), giving the stated uniform bound.

If $\hat{q}$ is obtained by normalizing a pre-activation vector q via $\hat{q} = q/\|q\|$, then

$$J(q) = \frac{1}{\|q\|}\big(I - \hat{q}\hat{q}^\top\big),$$

so $\|J(q)\|_{\text{op}} \leq 1/\|q\|$ wherever normalization is defined. Thus the gradient w.r.t. q is controlled by the spherical gradient and scales at most like $1/\|q\|$. □

Proof of Theorem ??.

Proof. By Lemma **??**,

$$k(x) = \frac{x^2}{C - 2x} = \int_0^\infty e^{-sC}\, x^2 e^{2sx}\, ds.$$

For each $s \geq 0$, the integrand $k_s(x) = x^2 e^{2sx}$ is PD because: (i) $x^2 = (\hat{q}^\top \hat{k})^2$ is PD as a degree-2 polynomial kernel restriction, (ii) for $s \geq 0$,

$$e^{2s(\hat{q}^\top \hat{k})} = \sum_{n=0}^{\infty} \frac{(2s)^n}{n!} (\hat{q}^\top \hat{k})^n$$

is a nonnegative linear combination of PD polynomial kernels, hence PD, and (iii) products of PD kernels are PD. Since the weight $e^{-sC} \geq 0$, the nonnegative mixture over s is PD by Lemma **??**. □