No More DeLuLu: Physics-Inspired Kernel Networks for Geometrically-Grounded Neural Computation

Anonymous Authors[1]

Abstract

We introduce the $\mathbb{E}$-product, a kernel operator combining quadratic alignment with inverse-square proximity. We prove it is a Mercer kernel—analytic, Lipschitz on bounded domains, and self-regularizing—admitting a unique RKHS embedding. Neural Matter Networks (NMNs) use $\mathbb{E}$-products as the sole nonlinearity, replacing conventional linear-activation-normalization blocks with a single geometrically-grounded operation. This architectural simplification preserves universal approximation while eliminating explicit activations and layer normalization. Empirically, NMN-based classifiers match linear baselines on MNIST while exhibiting bounded prototype evolution and superposition robustness. In language modeling, Aether-GPT2 achieves lower validation loss than GPT-2 (2.29 vs 2.43) with fewer components per layer. Our framework unifies kernel learning, gradient stability, and information geometry, establishing NMNs as a principled alternative to conventional neural architectures.

1. Introduction

Modern neural networks separate geometry from nonlinearity: dot products compute alignment, then activation functions like ReLU threshold the result (Goodfellow et al., 2016). This discards information—all negative activations become zero—requiring normalization layers and attention mechanisms to recover expressiveness (Ioffe & Szegedy, 2015; Vaswani et al., 2017).

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

We propose the $\mathbb{E}$-product, a neural operator that unifies alignment and proximity in a single computation:

$$\mathbb{E}(\mathbf{w}, \mathbf{x}) := \frac{\langle \mathbf{w}, \mathbf{x} \rangle^2}{\|\mathbf{w} - \mathbf{x}\|^2 + \varepsilon} \tag{1}$$

Inspired by inverse-square laws in physics, this operator creates a "potential well" around the weight vector $\mathbf{w}$: responses are high when inputs are both aligned *and* close, providing intrinsic non-linearity without thresholding. The $\mathbb{E}$-product is a Mercer kernel (Theorem 1) with universal approximation (Theorem 4), self-regulation (Proposition 1), and stable gradients (Proposition 2).

Using this kernel in primal form, we construct Neural-Matter Networks (NMNs)—networks where neurons interact through potential fields without requiring Gram matrix inversion. Our contributions: Our contributions span theory, architecture, and interpretability: the $\mathbb{E}$-product eliminates activation functions while maintaining Mercer kernel properties; NMNs reduce memory by 15-25% with infinite differentiability for physics-informed applications; and the geometric structure preserves spatial relationships (Theorems 2, 3), enabling principled analysis of learned representations.

2. Methodology: A Framework for Geometry-Aware Computation

The $\mathbb{E}$-product is formally defined as $\mathbb{E}(\mathbf{w}, \mathbf{x}) = \frac{(\mathbf{w}^\top \mathbf{x})^2}{\|\mathbf{w}-\mathbf{x}\|^2+\varepsilon}$. It exhibits a unique form of non-linearity. Unlike conventional activation functions (e.g., ReLU (Nair & Hinton, 2010), sigmoid) which are often applied as separate, somewhat heuristic, transformations to introduce non-linearity after a linear operation, the non-linearity in the $\mathbb{E}$-product arises directly from its mathematical structure. It is a function of the squared dot product (capturing alignment) and the inverse squared Euclidean distance (capturing proximity) between the weight vector $\mathbf{w}$ and the input vector $\mathbf{x}$. This formulation provides a rich, explainable non-linearity based on fundamental geometric and algebraic rela-

tionships, rather than an imposed, "artificial" nonlinear mapping. The interaction between the numerator and the denominator allows for complex responses that are inherently tied to the geometric interplay of the input vectors.

The $\mathbb{E}$-product creates a potential well around the weight vector $\mathbf{w}$, reflecting both alignment and proximity.

At initialization, this geometry also exhibits a favorable high-dimensional scaling behavior. Under standard assumptions of i.i.d. zero-mean, constant-variance coordinates for $\mathbf{x}, \mathbf{w} \in \mathbb{R}^d$, both the numerator $A(\mathbf{x}, \mathbf{w}) = (\mathbf{w}^\top \mathbf{x})^2$ and the denominator $r(\mathbf{x}, \mathbf{w}) = \|\mathbf{w} - \mathbf{x}\|^2$ grow linearly with dimension, while their ratio $K(\mathbf{x}, \mathbf{w}) = A/(r + \varepsilon)$ remains $\mathcal{O}(1)$ in expectation (Corollary 2). This self-normalizing $\mathcal{O}(1)$ scaling directly counters high-dimensional "saturation" concerns that arise for RBF kernels, whose values vanish exponentially with dimension.

As a Mercer kernel (Theorem 1), on every compact domain K the $\mathbb{E}$-product admits a unique RKHS (up to isometry) (Theorem 6) and inherits kernel method advantages. Importantly, this kernel is used in its primal form for weight prototype learning and optimization. Consequently, we do not use any Gram matrix, thereby bypassing the stability issues associated with its inversion in dual-form kernel regression (Schölkopf & Smola, 2002).

When the $\mathbb{E}$-product is applied to probability distributions in the simplex, its extremal values admit an information-geometric characterization:

Theorem 1 (Mercer property of the YAT product kernel)**.** *Let $\varepsilon > 0$ and define*

$$k_{\mathbb{E}}(x, w) \;=\; \frac{(x^\top w)^2}{\|x - w\|^2 + \varepsilon}, \qquad x, w \in \mathbb{R}^d.$$

Then for every compact set $K \subset \mathbb{R}^d$, the kernel $k_{\mathbb{E}}$ is symmetric, continuous, and positive definite on K. Consequently, $k_{\mathbb{E}}$ is a Mercer kernel on K.

Theorem 2 (Minimal Similarity and Statistical Orthogonality)**.** *Let $\mathbf{p}, \mathbf{q} \in \Delta^{n-1}$ be distinct distributions. Then $\mathbb{E}(\mathbf{p}, \mathbf{q}) = 0$ if and only if their supports are disjoint,* $\operatorname{supp}(\mathbf{p}) \cap \operatorname{supp}(\mathbf{q}) = \emptyset$*. In this case $D_{\mathrm{KL}}(\mathbf{p}\|\mathbf{q}) = \infty$ and the cross-entropy $H(\mathbf{p}, \mathbf{q})$ is infinite.*

Theorem 3 (Maximal (Singular) Similarity)**.** *Define the ε-dependent $\mathbb{E}$-product*

$$\mathbb{E}_\varepsilon(\mathbf{p}, \mathbf{q}) := \frac{(\mathbf{p}^\top \mathbf{q})^2}{\|\mathbf{p} - \mathbf{q}\|_2^2 + \varepsilon}.$$

Let $\varepsilon > 0$ and $\mathbf{p}, \mathbf{q} \in \Delta^{n-1}$. Then $\mathbb{E}_\varepsilon(\mathbf{p}, \mathbf{q})$ is finite for all $\mathbf{p}, \mathbf{q}$, and

$$\mathbb{E}_\varepsilon(\mathbf{p}, \mathbf{p}) = \frac{\|\mathbf{p}\|_2^4}{\varepsilon}.$$

In the singular limit $\varepsilon \to 0^+$, the self-similarity $\mathbb{E}_\varepsilon(\mathbf{p}, \mathbf{p})$ diverges.

Singular joint limit. *If $(\mathbf{q}_k)_{k \geq 1} \subset \Delta^{n-1}$ satisfies $\mathbf{q}_k \neq \mathbf{p}$ and $\|\mathbf{q}_k - \mathbf{p}\|_2 \to 0$, and if $\varepsilon_k \to 0^+$, then*

$$\mathbb{E}_{\varepsilon_k}(\mathbf{p}, \mathbf{q}_k) \to \infty.$$

Corollary 1 (Distributional Identity and KL)**.** *For distributions $\mathbf{p}, \mathbf{q} \in \Delta^{n-1}$, $D_{\mathrm{KL}}(\mathbf{p}\|\mathbf{q}) = 0$ if and only if $\mathbf{p} = \mathbf{q}$ (Gibbs' inequality (Cover & Thomas, 2006)). In this case the cross-entropy reduces to entropy: $H(\mathbf{p}, \mathbf{q}) = H(\mathbf{p})$.*

The $\mathbb{E}$-product creates a potential well around $\mathbf{w}$, where interaction strength diminishes with distance while preserving orientation sensitivity. The explicit gradient structure (Theorem 5) and stable gradient decay (Proposition 2) ensure that gradients vanish for distant inputs, providing natural localization. Input perturbation robustness (Proposition 4) guarantees bounded response changes on bounded domains (with constant controlled by ε). When applied to probability distributions, it connects geometry to information-theoretic extremes (Theorem 2, Theorem 3, Corollary 1).

2.1. Neural Matter Network (NMN) Layers

The $\mathbb{E}$-product serves as the foundation for Neural Matter Network layers, employing the non-linear, spatially-aware $\mathcal{K}_{\mathbb{E}}$-kernel as the primary interaction mechanism, replacing conventional linear projections ($\langle \mathbf{w}, \mathbf{x} \rangle$). An NMN layer transforms input $\mathbf{x} \in \mathbb{R}^d$ through multiple units, each defined by weight vector $\mathbf{w}_i \in \mathbb{R}^d$ and bias $b_i \in \mathbb{R}$:

$$h(\mathbf{x}) = \left(s \cdot \sum_{i=1}^{n} \mathcal{K}_{\mathbb{E}}(\mathbf{w}_i, \mathbf{x}, b_i) \right) = \left(s \cdot \sum_{i=1}^{n} \frac{(\mathbf{w}_i^\top \mathbf{x} + b_i)^2}{\|\mathbf{w}_i - \mathbf{x}\|^2 + \varepsilon} \right)$$

where s is a scaling factor and n denotes the number of units. Each unit responds based on both alignment and proximity to its learned weight vector, enabling universal function approximation (Theorem 4) as an intrinsic property of the $\mathcal{K}_{\mathbb{E}}$-kernel itself. The self-regulation property (Proposition 1) ensures that outputs remain bounded without requiring explicit normalization layers. Figure 1 illustrates the architectural simplification.

Theorem 4 (Universal approximation with $\mathbb{E}$-kernel)**.** *Let $\mathcal{X} \subset \mathbb{R}^d$ be a compact set. Define the class of*

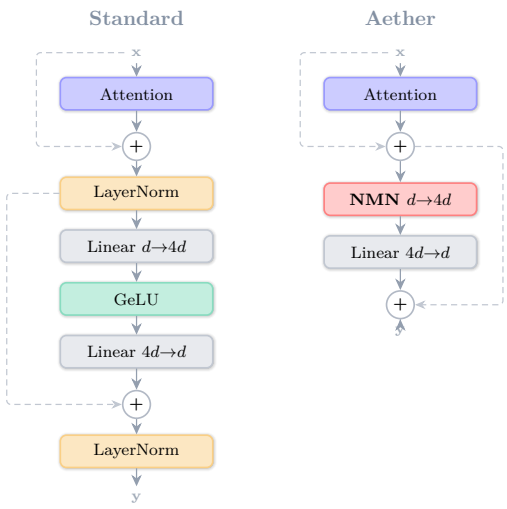

Figure 1. Comparison of Standard Transformer block (left) and Aether block (right). The NMN layer replaces Linear+GeLU, eliminating activation functions and all LayerNorm operations.

functions $\mathcal{F}$ realizable by the network as the linear span of the activation units:

$$\mathcal{F} = span\left\{ \frac{(\mathbf{x}\cdot\mathbf{w}+b)^2}{\|\mathbf{x}-\mathbf{w}\|^2+\varepsilon} \;\middle|\; \mathbf{w}\in\mathbb{R}^d, b\in\mathbb{R} \right\}$$

where $\varepsilon > 0$ is a fixed constant and b is the inner bias parameter. The set $\mathcal{F}$ is dense in $C(\mathcal{X})$ under the uniform norm.

Proposition 1 (Natural Self-Regulation)**.** *For any fixed $\mathbf{w}$ and unit direction $\mathbf{u}$, the $\mathbb{E}$-product output remains bounded as $k \to \infty$: $\lim_{k\to\infty} \mathbb{E}(\mathbf{w}, k\mathbf{u}) = \|\mathbf{w}\|^2 \cos^2\theta$, where θ is the angle between $\mathbf{w}$ and $\mathbf{u}$.*

Proposition 2 (Gradient Decay for Outliers)**.** *The gradient of the $\mathbb{E}$-product vanishes for distant inputs: $\lim_{\|\mathbf{x}\|\to\infty} \|\nabla_{\mathbf{x}}\mathbb{E}(\mathbf{w},\mathbf{x})\| = 0$.*

Theorem 5 (Gradient Direction)**.** *For two vectors $\mathbf{e}_i, \mathbf{e}_j$, the gradient is: $\nabla_{\mathbf{e}_i}\mathbb{E} = \frac{2\langle \mathbf{e}_i,\mathbf{e}_j\rangle}{\varepsilon+\|\mathbf{e}_i-\mathbf{e}_j\|^2}\left(\mathbf{e}_j - \frac{\langle \mathbf{e}_i,\mathbf{e}_j\rangle(\mathbf{e}_i-\mathbf{e}_j)}{\varepsilon+\|\mathbf{e}_i-\mathbf{e}_j\|^2}\right)$.*

Theorem 6 (RKHS Existence)**.** *For every compact set $K \subset \mathbb{R}^d$, the kernel $k_{\mathbb{E}}$ is positive definite on K (Theorem 1). Hence, by the Moore–Aronszajn theorem, there exists a unique RKHS $\mathcal{H}_K$ and feature map $\phi_K : K \to \mathcal{H}_K$ such that $k_{\mathbb{E}}(\mathbf{x},\mathbf{y}) = \langle \phi_K(\mathbf{x}), \phi_K(\mathbf{y})\rangle_{\mathcal{H}_K}$ for all $\mathbf{x},\mathbf{y} \in K$.*

Proposition 3 (Lipschitz Continuity)**.** *Fix $\varepsilon > 0$ and a weight vector $\mathbf{w}$ with $\|\mathbf{w}\|_2 \le 1$. Then the map $\mathbf{x} \mapsto \mathbb{E}(\mathbf{w},\mathbf{x})$ is Lipschitz continuous on the unit ball $\{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\|_2 \le 1\}$ with Lipschitz constant $L = 2/\varepsilon + 4/\varepsilon^2$.*

Lemma 1 (Analyticity)**.** *For $\varepsilon > 0$, the map $\mathbf{x} \mapsto \mathbb{E}(\mathbf{w},\mathbf{x})$ is real-analytic on $\mathbb{R}^d$ (infinitely differentiable).*

Proposition 4 (Input Perturbation Robustness)**.** *Fix $\varepsilon > 0$ and $\|\mathbf{w}\|_2 \le 1$. For any $\mathbf{x},\mathbf{x}'$ in the unit ball with $\|\mathbf{x}'-\mathbf{x}\|_2 \le \delta$,*

$$|\mathbb{E}(\mathbf{w},\mathbf{x}') - \mathbb{E}(\mathbf{w},\mathbf{x})| \le \left(\frac{2}{\varepsilon} + \frac{4}{\varepsilon^2}\right)\delta.$$

Corollary 2 (Dimensional Self-Normalization)**.** *Let $\mathbf{x},\mathbf{w} \in \mathbb{R}^d$ have i.i.d. zero-mean coordinates with $\mathrm{Var}(x_i) = \mathrm{Var}(w_i) = \sigma^2$ (constant in d), and assume in addition that:*

- *$\mathbf{x}$ and $\mathbf{w}$ are independent, and*
- *the coordinates are sub-Gaussian with parameter independent of d (e.g., Gaussian initialization), hence have finite fourth moments.*

Fix $\varepsilon > 0$. Then as $d \to \infty$,

$$\mathbb{E}[\mathbb{E}(\mathbf{w},\mathbf{x})] = \mathcal{O}(1).$$

Proposition 5 (Extremal Similarity on the Simplex)**.** *For $\mathbf{p},\mathbf{q} \in \Delta^{n-1}$: $\mathbb{E}(\mathbf{p},\mathbf{q}) = 0$ iff $\mathrm{supp}(\mathbf{p}) \cap \mathrm{supp}(\mathbf{q}) = \emptyset$, which implies $\mathrm{KL}(\mathbf{p}\|\mathbf{q}) = \infty$.*

Remark 1 (Optimal ε Scaling)**.** *For noisy inputs with $\mathbf{n} \sim \mathcal{N}(0,\sigma^2\mathbf{I})$, the stability constant should scale as $\varepsilon^* \propto d\sigma^2$ to maximize gradient signal-to-noise ratio.*

2.2. Architectural Implementation

Following the representer theorem (Schölkopf et al., 2001), optimal solutions in kernel methods lie in the span of kernel evaluations at training points. Since the $\mathbb{E}$-product is a Mercer kernel, composing multiple $\mathbb{E}$-layers without intervening linear projections would create a deep kernel that loses this representational guarantee. Our architecture therefore pairs each $\mathbb{E}$-kernel layer with a subsequent linear projection, preserving the kernel's theoretical properties while enabling depth. We also eliminate normalization layers entirely: the $\mathbb{E}$-product's self-regulation property (Proposition 1) provides intrinsic normalization, making explicit batch or layer normalization redundant and potentially harmful to gradient flow. The Lipschitz regularity (Proposition 3) and analyticity (Lemma 1) ensure stable training dynamics and infinite differentiability. All NMN-based layers use the adaptive scaling factor $s = \left(\frac{n}{\log(1+n)}\right)^{\alpha}$, where n is the number of units and α is a learnable parameter initialized at 1.

Computational Efficiency: The $\mathbb{E}$-product layer maintains $\Theta(Bnd)$ computational complexity identical to standard linear layers while providing 15-25%

memory reduction through elimination of activation storage. Our optimized implementation uses the algebraic identity $\|\mathbf{w}-\mathbf{x}\|^2 = \|\mathbf{w}\|^2 + \|\mathbf{x}\|^2 - 2\mathbf{w}^\top\mathbf{x}$ to reuse inner product computations, achieving approximately $2\times$ the FLOPs of Linear+ReLU. The approach offers natural numerical stability and becomes increasingly efficient at larger layer sizes, making it particularly suitable for large-scale applications.

3. Results and Discussion

We evaluate the $\mathbb{E}$-product on three tasks: XOR separability (demonstrating non-linearity), MNIST classification (prototype learning), and language modeling (Aether-GPT2).

3.1. XOR Separability with a Single Unit

The $\mathbb{E}$-product's inherent non-linearity enables solving non-linearly separable problems with a single unit. For XOR with inputs $(0,0) \to 0$, $(0,1) \to 1$, $(1,0) \to 1$, $(1,1) \to 0$, a single $\mathbb{E}$-product unit with $\mathbf{w} = [1,-1]^\top$ achieves perfect separation:

$\mathbf{x}$	$\mathbf{w}^\top\mathbf{x}$	$\mathcal{K}_{\mathbb{E}}(\mathbf{w},\mathbf{x})$	Class
$(0,0)$	0	0	0
$(0,1)$	-1	$1/(5+\varepsilon) > 0$	1
$(1,0)$	1	$1/(1+\varepsilon) > 0$	1
$(1,1)$	0	0	0

By definition (Eq. (1)), $\mathbf{w}^\top\mathbf{x} = 0$ implies $\mathcal{K}_{\mathbb{E}}(\mathbf{w},\mathbf{x}) = 0$ (in particular when $\mathbf{w} \perp \mathbf{x}$). See Appendix D for formal proof.

3.2. Decision Boundaries and Localization

Unlike linear neurons that induce unbounded hyperplane partitions, $\mathbb{E}$-product neurons generate localized decision surfaces around prototypes. This follows from self-regulation (Proposition 1): the response $\mathcal{K}_{\mathbb{E}}(\mathbf{w},\mathbf{x}) \to \|\mathbf{w}\|^2 \cos^2\theta$ as $\|\mathbf{x}\| \to \infty$.

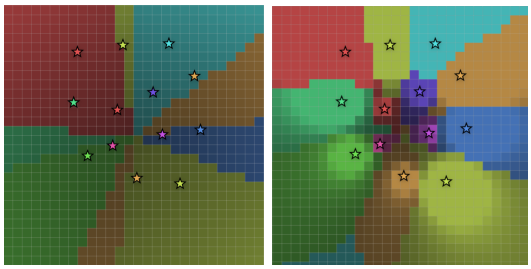

Figure 2. Decision boundaries in 2D: linear (left) creates unbounded half-spaces; $\mathbb{E}$-product (right) forms localized regions around prototypes (stars).

The extremal similarity results characterize boundary cases. Orthogonality ($\mathbf{w}^\top\mathbf{x} = 0$) yields $\mathcal{K}_{\mathbb{E}} = 0$ directly from Eq. (1). Identity ($\mathbf{w} = \mathbf{x}$) yields $\mathcal{K}_{\mathbb{E}}(\mathbf{w},\mathbf{w}) = \|\mathbf{w}\|_2^4/\varepsilon$ (Theorem 3 states this on the simplex; the same algebra holds in $\mathbb{R}^d$). Lipschitz continuity (Proposition 3) ensures smooth interpolation.

3.3. MNIST Classification

We compare a 10-neuron $\mathbb{E}$-product classifier against a linear baseline on MNIST (60k training, 10k test samples). Architecture: $C = 10$ prototypes $\mathbf{w}_i \in \mathbb{R}^{784}$. Training: Adam (lr=0.001), 5 epochs. Baseline: linear classifier $z_i = \mathbf{w}_i^\top\mathbf{x}$ with softmax.

Table 1. MNIST results (10-neuron classifier).

	Acc.	$\Delta\|\mathbf{w}\|$	α
Linear	92.08%	+13.8%	–
$\mathbb{E}$	**92.38%**	−4.5%	$1 \to 2.68$

Bounded Prototype Evolution. The self-regulation property (Proposition 1) predicts stable prototype magnitudes. Empirically, linear prototypes grow unboundedly (+13.8%), while $\mathbb{E}$-product prototypes contract slightly (−4.5%), confirming bounded response fields. The learnable scaling factor α (initialized at 1) increases to 2.68, amplifying bounded $\mathbb{E}$-responses for softmax discrimination.

Superposition Robustness. The squared numerator $(\mathbf{w}^\top\mathbf{x})^2$ creates approximate invariance under sign flip. Prototype inversion ($\mathbf{w} \to -\mathbf{w}$) yields:

	Original	Inverted
Linear	92.04%	0.01%
$\mathbb{E}$	92.18%	87.87%

For linear neurons, $(-\mathbf{w})^\top\mathbf{x} = -\mathbf{w}^\top\mathbf{x}$ flips the logit sign, causing catastrophic failure. For $\mathbb{E}$-product, the numerator invariance provides robustness.

Territorial Structure. Since the numerator is $(\mathbf{w}_i^\top\mathbf{w}_j)^2$, orthogonal prototypes satisfy $\mathbb{E}(\mathbf{w}_i,\mathbf{w}_j) = 0$. The $\mathbb{E}$-product develops heterogeneous structure: high similarity for morphologically similar digits (7-9), sharp boundaries elsewhere (Figure 4).

3.4. Language Modeling: Aether-GPT2

We train Aether-GPT2 (124M parameters) on 2.5B tokens from FineWeb. The architectural modifications follow Section 2.1: the MLP block is replaced by an

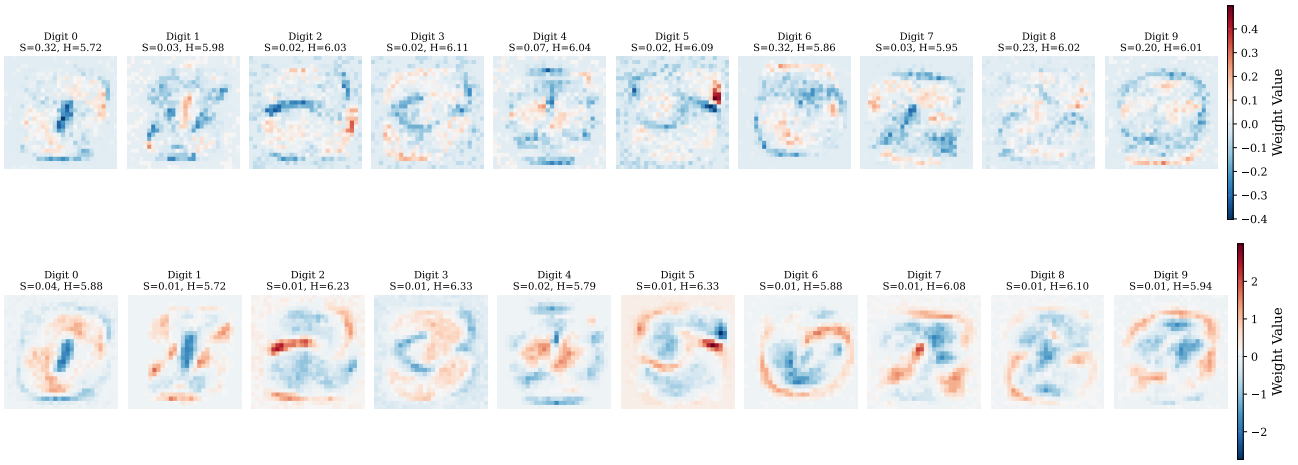

Figure 3. Learned prototypes: Linear (left) exhibits diffuse representations; Ɛ-product (right) shows sharp, geometrically coherent features.

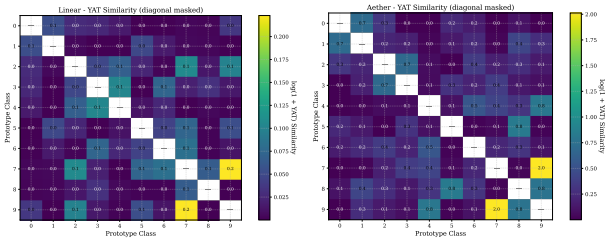

Figure 4. Pairwise Ɛ-similarity. Linear (left): uniform; Ɛ-product (right): heterogeneous territorial structure.

NMN layer (dim 3072, i.e., 4× hidden dim) followed by a linear projection, eliminating activation functions. All layer normalization is removed from attention skip connections to preserve geometric structure.

Table 2. GPT-2 vs Aether-GPT2 (2.5B tokens, identical hyperparameters).

	GPT-2	Aether
Activation	GeLU	**Ɛ**
LayerNorm	Yes	**No**
Val Loss (FP32)	2.43	**2.29**
Val Loss (BF16)	3.03	**2.69**
Improvement	–	**11.2%**
Throughput	138k	132k
Memory	–	−15–25%

Architectural Simplification. The self-regulation property (Proposition 1) and dimensional scaling (Corollary 2) eliminate the need for layer normalization. This yields 15–25% memory reduction by removing activation storage.

Mixed-Precision Stability. Under BF16 training, Aether-GPT2 achieves 2.69 validation loss versus 3.03 baseline—an 11.2% relative improvement. The bounded Ɛ-product response provides numerical stability without explicit normalization. The RKHS embedding (Theorem 6) ensures well-defined feature spaces.

See Appendix E.6 for detailed configuration and BF16 results.

4. Related Work

4.1. Kernel Methods and Neural Tangent Kernels

Kernel methods enable non-linear learning through implicit feature mappings (Schölkopf & Smola, 2002). SVMs (Cortes, 1995) and kernel PCA (Schölkopf et al., 1998) established the foundation, with Gaussian Processes (Williams & Rasmussen, 2006) extending to probabilistic inference. Scalability came through the Nyström method (Williams & Seeger, 2000) and Random Fourier Features (Rahimi & Recht, 2007).

The Neural Tangent Kernel (Jacot et al., 2018) bridges kernels and deep learning by characterizing infinite-width networks as linear models under gradient descent (Lee et al., 2019; Arora et al., 2019). Since the Ɛ-product is a valid Mercer kernel (Theorem 1), NTK theory extends to our framework (Proposition 6), enabling infinite-width analysis of geometric operators. The connection between SGD and kernel learning (Daniely, 2017; Li & Liang, 2018) further supports our approach.

Distance-based kernels (RBF) emphasize proximity;

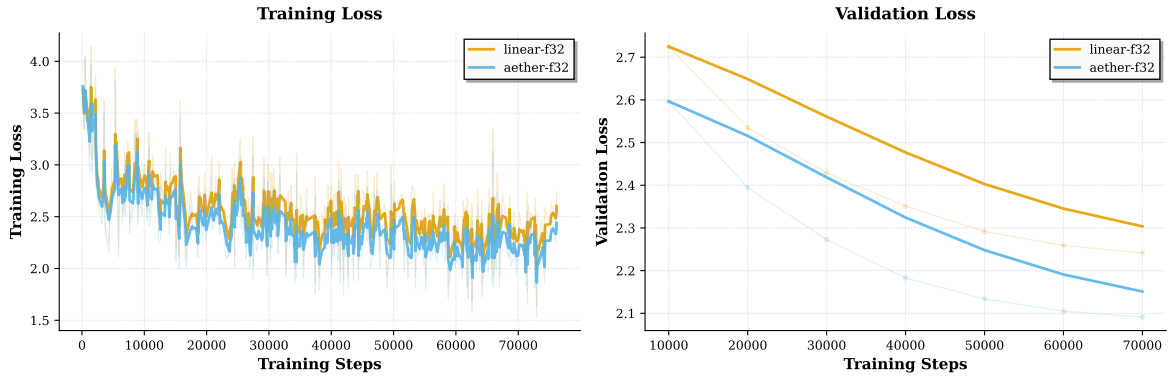

Figure 5. Training and validation loss curves. Aether-GPT2 consistently achieves lower loss throughout training.

polynomial kernels capture feature interactions. The 𝔼-product unifies both: the squared numerator provides polynomial-like alignment, while the inverse-square denominator gives RBF-like locality with intrinsic self-regularization.

Deep kernel learning (Wilson et al., 2016; Aitchison et al., 2021) combines neural networks with kernel flexibility, but operates in dual form requiring $O(n^2)$ Gram matrices. Our primal-form approach computes directly in feature space, avoiding this cost. Prior kernelized networks (Cho & Saul, 2009; Mairal et al., 2014) approximate kernels within linear-then-activate structures; the 𝔼-product is simultaneously the computational primitive and the kernel.

4.2. Alternative Neural Operators

Quadratic neurons (Fan et al., 2020; Liao et al., 2024) achieve non-linearity through polynomial forms but ignore geometric structure. Multiplicative interactions (Jayakumar et al., 2020) and gated linear units (Dauphin et al., 2016) introduce element-wise products yet retain activation dependence. SIREN (Sitzmann et al., 2020) and Fourier feature networks (Tancik et al., 2020) employ periodic activations for implicit representations.

The 𝔼-product differs fundamentally: it integrates alignment (squared dot product) and proximity (inverse distance) into a single operator, achieving non-linearity through geometric structure rather than functional composition—no activation functions required.

4.3. Geometric Foundations

The inverse-square law governs fundamental interactions across physics: gravitation (Newton, 1687), electrostatics (de Coulomb, 1785), and radiation (Gauss, 1835). This principle—intensity scaling inversely with squared distance—appears in engineering (signal propagation (Rappaport, 2002)) and information theory (Tanimoto similarity (Tanimoto, 1958)). Geometric deep learning (Bronstein et al., 2021) provides a unifying framework for exploiting such structure in neural architectures.

The 𝔼-product operationalizes this geometric principle for neural computation: interaction strength grows with alignment but decays with distance, providing a physics-inspired foundation for learning representations.

5. Conclusion

We introduced the 𝔼-product, a kernel operator that unifies alignment and proximity: $\mathcal{K}_{\mathbb{E}}(\mathbf{w}, \mathbf{x}) = (\mathbf{w}^\top \mathbf{x})^2/(\|\mathbf{w} - \mathbf{x}\|^2 + \varepsilon)$. We proved it is a valid Mercer kernel with analyticity, Lipschitz continuity on bounded domains, self-regulation, and gradient decay—properties that enable Neural Matter Networks to preserve universal approximation while eliminating explicit activations and normalization. Empirically, Aether-GPT2 achieves lower validation loss than GPT-2 (2.29 vs 2.43) with fewer components per layer. By grounding neural computation in physics-inspired geometry, this work offers a principled path toward simpler, more interpretable architectures.

License

licenses/agpl-3.0.en.html. By using this work, you agree to comply with the terms of the AGPL v3.0.

References

Aitchison, L., Yang, A., and Ober, S. W. Deep kernel processes. In *International Conference on Machine Learning*, pp. 130–140. PMLR, 2021.

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

Arora, S., Du, S. S., Hu, W., Li, Z., Salakhutdinov, R., and Wang, R. On exact computation with an infinitely wide neural net. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Bronstein, M. M., Bruna, J., Cohen, T., and Veličković, P. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges, 2021. URL https://arxiv.org/abs/2104.13478.

Cho, Y. and Saul, L. Kernel methods for deep learning. In Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C., and Culotta, A. (eds.), *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc., 2009. URL https://proceedings.neurips.cc/paper_files/paper/2009/file/5751ec3e9a4feab575962e78e006250d-Paper.pdf.

Cortes, C. Support-vector networks. *Machine Learning*, 1995.

Cover, T. M. and Thomas, J. A. *Elements of Information Theory*. Wiley-Interscience, 2006.

Daniely, A. Sgd learns the conjugate kernel class of the network. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Dauphin, Y. N., Fan, A., Auli, M., and Grangier, D. Language modeling with gated convolutional networks. *CoRR*, abs/1612.08083, 2016. URL http://arxiv.org/abs/1612.08083.

de Coulomb, C.-A. Premier mémoire sur l'électricité et le magnétisme. *Histoire de l'Académie Royale des Sciences*, pp. 1–31, 1785. in French.

Fan, F., Xiong, J., and Wang, G. Universal approximation with quadratic deep networks. *Neural Networks*, 124:383–392, 2020. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2020.01.007. URL https://www.sciencedirect.com/science/article/pii/S0893608020300095.

Folland, G. B. *Real Analysis: Modern Techniques and Their Applications*. Wiley, 2nd edition, 1999.

Gauss, C. F. *Allgemeine Lehrsätze in Beziehung auf die im verkehrten Verhältniss des Quadrats der Entfernung wirkenden Anziehungs- und Abstossungskräfte*. Dietrich, Göttingen, 1835.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep learning*, volume 1. MIT Press, 2016.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015. URL https://arxiv.org/abs/1502.03167.

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Jayakumar, S. M., Czarnecki, W. M., Menick, J., Schwarz, J., Rae, J., Osindero, S., Teh, Y. W., Harley, T., and Pascanu, R. Multiplicative interactions and where to find them. 2020.

Lee, J., Xiao, L., Schoenholz, S. S., Bahri, Y., Novak, R., Sohl-Dickstein, J., and Pennington, J. Wide neural networks of any depth evolve as linear models under gradient descent. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Li, Y. and Liang, Y. Learning overparameterized neural networks via stochastic gradient descent on structured data. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Liao, J.-X., Hou, B.-J., Dong, H.-C., Zhang, H., Zhang, X., Sun, J., Zhang, S., and Fan, F.-L. Quadratic neuron-empowered heterogeneous autoencoder for unsupervised anomaly detection. *IEEE Transactions on Artificial Intelligence*, 5(9): 4723–4737, 2024. doi: 10.1109/TAI.2024.3394795.

Mairal, J., Koniusz, P., Harchaoui, Z., and Schmid, C. Convolutional kernel networks, 2014. URL https://arxiv.org/abs/1406.3332.

Mercer, J. Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society of London. Series A*, 209:415–446, 1909.

Micchelli, C. A., Xu, Y., and Zhang, H. Universal kernels. *Journal of Machine Learning Research*, 7 (12), 2006.

Nair, V. and Hinton, G. E. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814, 2010.

Newton, I. *Philosophiæ Naturalis Principia Mathematica*. S. Pepys, London, 1687.

Rahimi, A. and Recht, B. Random features for large-scale kernel machines. *Advances in neural information processing systems*, 20, 2007.

Rappaport, T. S. *Wireless Communications: Principles and Practice*. Prentice Hall, Upper Saddle River, NJ, 2 edition, 2002.

Rudin, W. *Functional Analysis*. McGraw-Hill, 1991.

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

Schölkopf, B. and Smola, A. J. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

Schölkopf, B., Smola, A., and Müller, K.-R. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319, 1998.

Schölkopf, B., Herbrich, R., and Smola, A. J. A generalized representer theorem. In *Computational Learning Theory*, pp. 416–426. Springer, 2001.

Sitzmann, V., Martel, J., Bergman, A., Lindell, D., and Wetzstein, G. Implicit neural representations with periodic activation functions. volume 33, pp. 7462–7473, 2020.

Steinwart, I. On the influence of the kernel on the consistency of support vector machines. *Journal of Machine Learning Research*, 2:67–93, 2001.

Tancik, M., Srinivasan, P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J., and Ng, R. Fourier features let networks learn high frequency functions in low dimensional domains. volume 33, pp. 7537–7547, 2020.

Tanimoto, T. T. An elementary mathematical theory of classification and prediction. 1958.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Wendland, H. *Scattered Data Approximation*. Cambridge University Press, 2005.

Williams, C. and Seeger, M. Using the nyström method to speed up kernel machines. *Advances in neural information processing systems*, 13, 2000.

Williams, C. K. and Rasmussen, C. E. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

Wilson, A. G., Hu, Z., Salakhutdinov, R., and Xing, E. P. Deep kernel learning. In *Artificial intelligence and statistics*, pp. 370–378. PMLR, 2016.

A. Appendix

B. Squashing Functions for Non-Negative Scores

The non-negative nature of $\mathbb{E}$-product scores necessitates specialized normalization functions. We categorize these into *competitive* (vector-normalizing) and *individualistic* (element-wise) functions.

B.1. Competitive Normalization

Competitive functions induce coupling between dimensions, interpreting scores as relative strengths within a distribution.

Softermax. A generalized normalization function for non-negative scores $\mathbf{x} \in \mathbb{R}^d_{\geq 0}$:

$$\text{softermax}_n(x_k, \{\mathbf{x}\}) = \frac{x_k^n}{\epsilon + \sum_{i=1}^d x_i^n}, \tag{2}$$

where $n > 0$ controls the distribution sharpness (analogous to inverse temperature) and $\epsilon > 0$ ensures numerical stability and prevents division by zero for sparse inputs. Unlike softmax, this formulation avoids exponential terms, improving numerical stability for large input magnitudes.

B.2. Individualistic Squashing

Individualistic functions map scores to bounded intervals element-wise, preserving independence.

Soft-Sigmoid. Maps $x \in [0, \infty)$ to $[0, 1)$:

$$\sigma_n(x) = \frac{x^n}{1 + x^n}. \tag{3}$$

This algebraic sigmoid provides a heavy-tailed alternative to the logistic sigmoid, with polynomial rather than exponential saturation.

Soft-Tanh. Maps $x \in [0, \infty)$ to $[-1, 1)$:

$$\tau_n(x) = \frac{x^n - 1}{x^n + 1}. \tag{4}$$

This corresponds to a rescaled soft-sigmoid: $\tau_n(x) = 2\sigma_n(x) - 1$. The parameter n acts as a gain factor, controlling the steepness of the transition from the distinct states -1 (orthogonality/dissimilarity) to $+1$ (alignment/similarity).

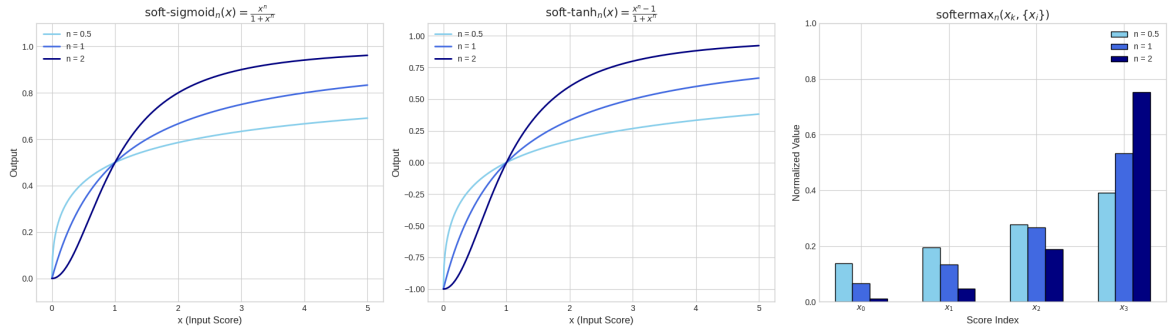

Figure 6. Algebraic squashing functions for non-negative $\mathbb{E}$-product scores. These offer bounded, monotonic mappings without exponential saturation.

B.3. Mathematical Preliminaries

This section collects the key mathematical tools and terminology used throughout the paper.

B.3.1. Kernel Terminology

Definition 1 (Kernel)**.** *A* kernel *is a function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ that measures similarity between two inputs (Schölkopf & Smola, 2002).*

Definition 2 (Gram Matrix)**.** *Given points $x_1, \ldots, x_n \in \mathcal{X}$, the* Gram matrix *of k is $K \in \mathbb{R}^{n \times n}$ with entries $K_{ij} = k(x_i, x_j)$.*

Definition 3 (Positive Definite Kernel)**.** *A symmetric kernel k is* positive definite (PD) *if its Gram matrix is positive semidefinite for every finite set of points:*

$$\sum_{i,j=1}^{n} a_i a_j \, k(x_i, x_j) \geq 0 \quad \textit{for all } a \in \mathbb{R}^n.$$

Definition 4 (Feature Map and RKHS)**.** *A* feature map *is a map $\Phi : \mathcal{X} \to \mathcal{H}$ such that $k(x, y) = \langle \Phi(x), \Phi(y) \rangle_{\mathcal{H}}$. The* reproducing kernel Hilbert space (RKHS) *of a PD kernel is the Hilbert space where the kernel becomes an inner product (Aronszajn, 1950).*

B.3.2. Closure Properties of PD Kernels

Theorem 7 (PD Closure Properties)**.** *If k_1, k_2 are positive definite kernels on $\mathcal{X}$, then (Schölkopf & Smola, 2002):*

1. *$k_1 + k_2$ is PD (closure under addition)*
2. *$w \cdot k_1$ is PD for $w \geq 0$ (closure under nonnegative scaling)*
3. *$k_1 \cdot k_2$ is PD (Schur product theorem)*
4. *$\int k_s \, w(s) \, ds$ is PD for $w(s) \geq 0$ (closure under nonnegative mixtures)*

Context: These closure properties are used in the Mercer kernel proof (Theorem 1) to show that the product of the polynomial kernel and inverse multiquadric is PD.

B.3.3. Laplace Transform Identity

Theorem 8 (Laplace Identity for $1/y$)**.** *For any $y > 0$:*

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

Context: This identity converts the rational term $1/(\|\mathbf{x} - \mathbf{w}\|^2 + \varepsilon)$ into an integral over exponentials, enabling the inverse multiquadric to be expressed as a nonnegative mixture of Gaussian kernels.

B.3.4. Complete Monotonicity and Bernstein Representation

Theorem 9 (Bernstein's Theorem)**.** *A function $f : [0, \infty) \to \mathbb{R}$ is completely monotonic (i.e., $(-1)^n f^{(n)}(y) \geq 0$ for all $n \geq 0$) if and only if it is the Laplace transform of a non-negative measure (Schilling et al., 2012):*

$$f(t) = \int_0^\infty e^{-ts} \, d\mu(s), \quad \mu \geq 0.$$

Context: The function $1/y$ is completely monotone, so it admits a nonnegative exponential-mixture representation. This justifies the decomposition used in the Mercer proof.

B.3.5. Integral Exchange (Tonelli-Fubini)

Theorem 10 (Tonelli-Fubini)**.** *For σ-finite measure spaces and measurable $f \geq 0$ (Folland, 1999):*

$$\int_{X \times Y} f \, d(\mu \times \nu) = \int_X \left(\int_Y f \, d\nu \right) d\mu = \int_Y \left(\int_X f \, d\mu \right) d\nu.$$

Context: Justifies exchanging integrals and sums in PD kernel proofs: "integral of PD kernels is PD" requires moving the integral outside the quadratic form.

B.3.6. Information Theory

Theorem 11 (Gibbs' Inequality)**.** *For probability distributions P, Q (Cover & Thomas, 2006):*

$$D_{KL}(P\|Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)} \geq 0,$$

with equality iff $P = Q$.

Context: Referenced in Corollary 1 connecting distributional identity to vanishing KL divergence.

B.3.7. Harmonic Analysis

Theorem 12 (Bochner's Theorem)**.** *A continuous function $k : \mathbb{R}^d \to \mathbb{C}$ is positive definite and translation-invariant iff it is the Fourier transform of a finite non-negative measure (Rudin, 1991).*

Theorem 13 (Hahn-Banach Density Criterion)**.** *A linear subspace $M \subset V$ is dense in V iff every continuous linear functional vanishing on M vanishes on V (Rudin, 1991).*

Context: Used in the universal approximation proof (Theorem 4).

C. Proofs of Main Theorems

This section provides the proofs for theorems stated in the main body.

C.1. Proof of Theorem 5 (Gradient Direction)

Let $s = \langle \mathbf{e}_i, \mathbf{e}_j \rangle$ and $D = \varepsilon + \|\mathbf{e}_i - \mathbf{e}_j\|^2$. Then $\mathbb{E} = s^2/D$.

Using the quotient rule:

$$\nabla_{\mathbf{e}_i} \mathbb{E} = \frac{2s \cdot \nabla_{\mathbf{e}_i} s \cdot D - s^2 \cdot \nabla_{\mathbf{e}_i} D}{D^2}.$$

We have $\nabla_{\mathbf{e}_i} s = \mathbf{e}_j$ and $\nabla_{\mathbf{e}_i} D = 2(\mathbf{e}_i - \mathbf{e}_j)$. Substituting:

$$\nabla_{\mathbf{e}_i} \mathbb{E} = \frac{2s\mathbf{e}_j \cdot D - 2s^2(\mathbf{e}_i - \mathbf{e}_j)}{D^2} = \frac{2s}{D}\left(\mathbf{e}_j - \frac{s(\mathbf{e}_i - \mathbf{e}_j)}{D}\right). \quad \square$$

C.2. Proof of Proposition 3 (Lipschitz Continuity)

Fix $\varepsilon > 0$ and assume $\|\mathbf{x}\|_2 \leq 1$ and $\|\mathbf{w}\|_2 \leq 1$. Let $s = \langle \mathbf{x}, \mathbf{w} \rangle$ and $D = \varepsilon + \|\mathbf{x} - \mathbf{w}\|_2^2$. Then $D \geq \varepsilon$. From the gradient formula (Theorem 5):

$$\|\nabla_{\mathbf{x}} \mathbb{E}\| \leq \frac{2|s|}{D}\left(\|\mathbf{w}\|_2 + \frac{|s| \cdot \|\mathbf{x} - \mathbf{w}\|_2}{D}\right).$$

For $\|\mathbf{x}\|_2, \|\mathbf{w}\|_2 \leq 1$, we have $|s| = |\langle \mathbf{x}, \mathbf{w} \rangle| \leq 1$ and $\|\mathbf{x} - \mathbf{w}\|_2 \leq 2$, so:

$$\|\nabla_{\mathbf{x}} \mathbb{E}\| \leq \frac{2}{\varepsilon}\left(1 + \frac{2}{\varepsilon}\right) = \frac{2}{\varepsilon} + \frac{4}{\varepsilon^2}. \quad \square$$

C.3. Proof of Proposition 5 (Extremal Similarity)

(1) The numerator $\langle \mathbf{p}, \mathbf{q} \rangle^2 = \left(\sum_i p_i q_i\right)^2 = 0$ if and only if all terms $p_i q_i = 0$, which occurs precisely when supports are disjoint.

(2) If $\text{supp}(\mathbf{p}) \cap \text{supp}(\mathbf{q}) = \emptyset$, there exists i with $p_i > 0$ and $q_i = 0$, making $\text{KL}(\mathbf{p}\|\mathbf{q}) = \sum_i p_i \log(p_i/q_i) = \infty$.

Cross-entropy (used in Theorem 2). Under the same condition, the cross-entropy

$$H(\mathbf{p},\mathbf{q}) = -\sum_i p_i \log q_i$$

contains a term with $p_i > 0$ and $q_i = 0$, hence $-\log q_i = -\log 0 = +\infty$ and therefore $H(\mathbf{p},\mathbf{q}) = \infty$ under the standard convention $\log 0 = -\infty$. □

C.4. Proof of Theorem 2 (Minimal Similarity and Statistical Orthogonality)

Let $\mathbf{p},\mathbf{q} \in \Delta^{n-1}$ and $\varepsilon > 0$. Since $\|\mathbf{p}-\mathbf{q}\|_2^2 + \varepsilon > 0$, we have $\mathbb{E}(\mathbf{p},\mathbf{q}) = 0$ if and only if $\mathbf{p}^\top\mathbf{q} = 0$. On the simplex, $p_i q_i \geq 0$ for all i, so $\sum_i p_i q_i = 0$ holds if and only if $p_i q_i = 0$ for all i, i.e. $\mathrm{supp}(\mathbf{p}) \cap \mathrm{supp}(\mathbf{q}) = \emptyset$. In this case $D_{\mathrm{KL}}(\mathbf{p}\|\mathbf{q}) = \infty$ by the argument in Proposition 5, and $H(\mathbf{p},\mathbf{q}) = \infty$ by the cross-entropy note above. □

C.5. Proof of Theorem 3 (Maximal (Singular) Similarity)

Fix $\varepsilon > 0$ and define $\mathbb{E}_\varepsilon(\mathbf{p},\mathbf{q}) := \frac{(\mathbf{p}^\top\mathbf{q})^2}{\|\mathbf{p}-\mathbf{q}\|_2^2+\varepsilon}$. For any $\mathbf{p},\mathbf{q} \in \Delta^{n-1}$, the denominator is at least ε, hence $\mathbb{E}_\varepsilon(\mathbf{p},\mathbf{q})$ is finite. If $\mathbf{q} = \mathbf{p}$, then $\|\mathbf{p}-\mathbf{q}\|_2^2 = 0$ and

$$\mathbb{E}_\varepsilon(\mathbf{p},\mathbf{p}) = \frac{(\mathbf{p}^\top\mathbf{p})^2}{\varepsilon} = \frac{\|\mathbf{p}\|_2^4}{\varepsilon},$$

so $\mathbb{E}_\varepsilon(\mathbf{p},\mathbf{p}) \to \infty$ as $\varepsilon \to 0^+$.

Singular joint limit. Let $(\mathbf{q}_k)_{k\geq 1} \subset \Delta^{n-1}$ satisfy $\mathbf{q}_k \neq \mathbf{p}$ and $\|\mathbf{q}_k - \mathbf{p}\|_2 \to 0$, and let $\varepsilon_k \to 0^+$. Then $(\mathbf{p}^\top\mathbf{q}_k)^2 \to (\mathbf{p}^\top\mathbf{p})^2 = \|\mathbf{p}\|_2^4 > 0$ while $\|\mathbf{p}-\mathbf{q}_k\|_2^2 + \varepsilon_k \to 0$. Therefore $\mathbb{E}_{\varepsilon_k}(\mathbf{p},\mathbf{q}_k) \to \infty$. □

C.6. Proof of Corollary 1 (Distributional Identity and KL)

By Gibbs' inequality (see e.g. Cover & Thomas (2006)), $D_{\mathrm{KL}}(\mathbf{p}\|\mathbf{q}) \geq 0$ with equality if and only if $\mathbf{p} = \mathbf{q}$. When $\mathbf{p} = \mathbf{q}$, the cross-entropy satisfies $H(\mathbf{p},\mathbf{q}) = H(\mathbf{p})$. □

C.7. Proof of Proposition 1 (Self-Regulation)

For $\mathbf{x} = k\mathbf{u}$:

$$\mathbb{E}(\mathbf{w}, k\mathbf{u}) = \frac{(k\mathbf{w}^\top\mathbf{u})^2}{\|\mathbf{w}-k\mathbf{u}\|^2+\varepsilon} = \frac{k^2(\mathbf{w}^\top\mathbf{u})^2}{\|\mathbf{w}\|^2 - 2k\mathbf{w}^\top\mathbf{u} + k^2 + \varepsilon}.$$

Dividing numerator and denominator by k^2:

$$\mathbb{E}(\mathbf{w}, k\mathbf{u}) = \frac{(\mathbf{w}^\top\mathbf{u})^2}{\|\mathbf{w}\|^2/k^2 - 2\mathbf{w}^\top\mathbf{u}/k + 1 + \varepsilon/k^2} \to (\mathbf{w}^\top\mathbf{u})^2 = \|\mathbf{w}\|^2\cos^2\theta. \quad \square$$

C.8. Proof of Proposition 2 (Gradient Decay)

From Theorem 5, with $s = \langle\mathbf{x},\mathbf{w}\rangle$ and $D = \varepsilon + \|\mathbf{x}-\mathbf{w}\|_2^2$,

$$\|\nabla_{\mathbf{x}}\mathbb{E}\| \leq \frac{2|s|}{D}\left(\|\mathbf{w}\|_2 + \frac{|s|\,\|\mathbf{x}-\mathbf{w}\|_2}{D}\right).$$

As $\|\mathbf{x}\| \to \infty$ with fixed $\mathbf{w}$, we have $|s| = O(\|\mathbf{x}\|)$, $\|\mathbf{x}-\mathbf{w}\|_2 = O(\|\mathbf{x}\|)$, and $D = O(\|\mathbf{x}\|^2)$, hence the right-hand side is $O(1/\|\mathbf{x}\|) \to 0$. □

C.9. Proof of Corollary 2 (Dimensional Scaling)

Assume $\mathbf{x},\mathbf{w} \in \mathbb{R}^d$ have i.i.d. zero-mean coordinates with $\mathrm{Var}(x_i) = \mathrm{Var}(w_i) = \sigma^2$ independent of d, and assume in addition that $\mathbf{x}$ and $\mathbf{w}$ are independent. Then

$$\mathbb{E}\big[(\mathbf{w}^\top\mathbf{x})^2\big] = \sum_{i=1}^d \mathbb{E}[w_i^2 x_i^2] = d\,\sigma^4 = \Theta(d),$$

since cross-terms vanish by independence and zero mean. Moreover,

$$\mathbb{E}\big[\|\mathbf{w}-\mathbf{x}\|_2^2\big] = \mathbb{E}\big[\|\mathbf{w}\|_2^2\big] + \mathbb{E}\big[\|\mathbf{x}\|_2^2\big] - 2\mathbb{E}[\mathbf{w}^\top\mathbf{x}] = 2d\sigma^2 = \Theta(d).$$

The above shows the numerator and denominator scale linearly in d in expectation, but to control the expectation of the ratio we use Cauchy–Schwarz. Assume in addition the coordinates are sub-Gaussian with parameter independent of d (hence have finite fourth moments). Let $U := (\mathbf{w}^\top\mathbf{x})^2$ and $V := \|\mathbf{w}-\mathbf{x}\|_2^2$, so $\mathbb{E} = U/(V+\varepsilon)$. Then $\mathbb{E}[U^2] = \mathbb{E}[(\mathbf{w}^\top\mathbf{x})^4] = \mathcal{O}(d^2)$.

Bounding the inverse square moment. Let $z_i := w_i - x_i$. By independence and sub-Gaussianity, $(z_i)_{i=1}^d$ are i.i.d. mean-zero sub-Gaussian with $\mathbb{E}[z_i^2] = 2\sigma^2$, and

$$V = \sum_{i=1}^{d} z_i^2.$$

Since $z_i^2 - \mathbb{E}[z_i^2]$ is sub-exponential, Bernstein's inequality implies there exists $c > 0$ (independent of d) such that

$$\mathbb{P}\big(V \le \tfrac{1}{2}\mathbb{E}[V]\big) = \mathbb{P}\big(V \le d\sigma^2\big) \le e^{-cd}.$$

Therefore, for fixed $\varepsilon > 0$,

$$\mathbb{E}\big[(V+\varepsilon)^{-2}\big] = \mathbb{E}\big[(V+\varepsilon)^{-2}\mathbf{1}_{\{V\ge d\sigma^2\}}\big] \;+\; \mathbb{E}\big[(V+\varepsilon)^{-2}\mathbf{1}_{\{V<d\sigma^2\}}\big] \le \frac{1}{(d\sigma^2)^2} + \frac{1}{\varepsilon^2}e^{-cd} = \mathcal{O}(d^{-2}).$$

Therefore

$$\mathbb{E}[\mathbb{E}] = \mathbb{E}\left[\frac{U}{V+\varepsilon}\right] \le \sqrt{\mathbb{E}[U^2]}\,\sqrt{\mathbb{E}[(V+\varepsilon)^{-2}]} = \mathcal{O}(1),$$

as $d \to \infty$. □

C.10. Proof of Theorem 6 (RKHS Existence)

Since $k_{\mathbb{E}}$ is positive semi-definite on every compact set $K \subset \mathbb{R}^d$ (Theorem 1), the Moore–Aronszajn theorem (Aronszajn, 1950) guarantees the existence of a (unique up to isometry) RKHS $\mathcal{H}_K$ and feature map $\phi_K : K \to \mathcal{H}_K$ such that $k_{\mathbb{E}}(\mathbf{x},\mathbf{y}) = \langle\phi_K(\mathbf{x}), \phi_K(\mathbf{y})\rangle_{\mathcal{H}_K}$ for all $\mathbf{x},\mathbf{y} \in K$. □

C.11. Proof of Proposition 4 (Input Robustness)

Assume $\|\mathbf{w}\|_2 \le 1$ and $\mathbf{x},\mathbf{x}'$ lie in the unit ball. By the mean value theorem on the line segment between $\mathbf{x}$ and $\mathbf{x}'$ (which stays in the unit ball by convexity) and Lipschitz continuity (Proposition 3):

$$|\mathbb{E}(\mathbf{w},\mathbf{x}') - \mathbb{E}(\mathbf{w},\mathbf{x})| \le L\cdot\|\boldsymbol{\delta}\| = \left(\frac{2}{\varepsilon} + \frac{4}{\varepsilon^2}\right)\delta.$$

In particular, for fixed ε this is $O(\delta)$ (and for small ε the constant scales as $O(\varepsilon^{-2})$). □

C.12. Proof of Lemma 1 (Analyticity)

The $\mathbb{E}$-product is a ratio of polynomials where the denominator $\|\mathbf{w}-\mathbf{x}\|^2 + \varepsilon \ge \varepsilon > 0$ is bounded away from zero. Ratios of polynomials with non-vanishing denominators are real-analytic. □

C.13. Justification for Remark 1 (Optimal ε)

For input data with additive noise $\mathbf{n} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$, the noise contribution to $\|\mathbf{w}-\mathbf{x}\|^2$ is $O(d\sigma^2)$ in expectation. Setting $\varepsilon^* \propto d\sigma^2$ ensures the noise floor matches the stability constant, maximizing the signal-to-noise ratio of gradients.

C.14. Topological Properties of Neural Activation Functions

Theorem 14 (Non-Homeomorphism of Standard Activations)**.** *Let $T : \mathbb{R}^d \to \mathbb{R}^m$ be defined by $T(\mathbf{x}) = \phi(W\mathbf{x}+\mathbf{b})$ where $W \in \mathbb{R}^{m\times d}$, $\mathbf{b} \in \mathbb{R}^m$, and ϕ is an element-wise activation function. Then:*

1. *If $\phi = ReLU$ and there exist two distinct inputs $\mathbf{x}_1 \neq \mathbf{x}_2$ such that $W\mathbf{x}_1 + \mathbf{b} \leq 0$ and $W\mathbf{x}_2 + \mathbf{b} \leq 0$ coordinatewise, then T is not injective.*
2. *If $\phi \in \{sigmoid, \tanh\}$, then T fails to be bi-Lipschitz. For inputs $\mathbf{z}_1, \mathbf{z}_2$ in the saturation regime ($|z| \to \infty$), $\|\phi(\mathbf{z}_1) - \phi(\mathbf{z}_2)\| \to 0$ regardless of $\|\mathbf{z}_1 - \mathbf{z}_2\|$.*
3. *In particular, in the ReLU case above, T is not a homeomorphism onto its image (since it is not injective). In the sigmoid/tanh case, T can be a homeomorphism onto its image when W is injective, but it is not a bi-Lipschitz embedding (metric structure can be arbitrarily compressed in saturation).*

Proof. (1) *ReLU non-injectivity under clipping*: For $\phi(z) = \max(0, z)$, if $(W\mathbf{x} + \mathbf{b})_i \leq 0$ then the i-th output coordinate equals 0. If there exist $\mathbf{x}_1 \neq \mathbf{x}_2$ such that $W\mathbf{x}_1 + \mathbf{b} \leq 0$ and $W\mathbf{x}_2 + \mathbf{b} \leq 0$ coordinatewise, then $T(\mathbf{x}_1) = T(\mathbf{x}_2) = \mathbf{0}$, hence T is not injective.

(2) *Sigmoid/tanh saturation*: Consider $\sigma(z) = 1/(1 + e^{-z})$. For $z \to +\infty$, $\sigma(z) \to 1$ with $|\sigma(z_1) - \sigma(z_2)| \leq e^{-\min(z_1,z_2)}|z_1 - z_2|$ for large z_1, z_2. Thus the Lipschitz constant in the saturation regime approaches zero—distances are compressed. Similarly for $\tanh(z) = 2\sigma(2z) - 1$.

(3) *Topological vs metric structure*: By (1), ReLU can fail injectivity, hence cannot be a homeomorphism onto its image. By (2), sigmoid/tanh fail bi-Lipschitz (no uniform lower Lipschitz bound), which is a metric distortion statement and does not by itself preclude homeomorphism when W is injective. □

Remark 2 (Information loss under non-invertible activations (informal))**.** *By the data processing inequality, representations obtained by applying a deterministic non-invertible nonlinearity (e.g. ReLU clipping) cannot increase information about the input. Making this statement fully rigorous requires specifying a probabilistic model (often with additive noise) to avoid pathologies of mutual information for continuous deterministic transforms.*

Remark 3 (Contrast with 𝔼-Product)**.** *The 𝔼-product achieves non-linearity through its rational structure $\mathcal{K}_{\mathbb{E}}(\mathbf{w}, \mathbf{x}) = \frac{(\mathbf{w}^\top \mathbf{x})^2}{\|\mathbf{w}-\mathbf{x}\|^2+\varepsilon}$ without collapsing regions. For $\varepsilon > 0$, the mapping is:*

- *Real-analytic (Lemma 1)*
- *Lipschitz on bounded sets (Proposition 3)*
- *Self-regulating with bounded output (Proposition 1)*

The denominator $\varepsilon > 0$ prevents singularities, while the squared numerator provides non-linearity without hard thresholding.

C.15. Proof of Theorem 1 (Mercer Property)

Fix $\varepsilon > 0$ and let $K \subset \mathbb{R}^d$ be compact.

Step 1: Integral representation. For any $a > 0$,

$$\frac{1}{a} = \int_0^\infty e^{-ta}\, dt.$$

Since $\|x - w\|^2 + \varepsilon > 0$ for all $x, w \in \mathbb{R}^d$, we apply this identity with $a = \|x - w\|^2 + \varepsilon$ to obtain

$$\frac{1}{\|x - w\|^2 + \varepsilon} = \int_0^\infty e^{-t\varepsilon} e^{-t\|x-w\|^2}\, dt. \tag{5}$$

Multiplying by $(x^\top w)^2$ yields

$$k_{\mathbb{E}}(x, w) = \int_0^\infty (x^\top w)^2 e^{-t\varepsilon} e^{-t\|x-w\|^2}\, dt. \tag{6}$$

Step 2: Positive definite components. For each $t > 0$, the Gaussian kernel

$$g_t(x, w) = e^{-t\|x-w\|^2}$$

is positive definite on $\mathbb{R}^d$ (Schölkopf & Smola, 2002). The polynomial kernel

$$p(x, w) = (x^\top w)^2$$

is also positive definite, since

$$(x^\top w)^2 = \langle x \otimes x,\ w \otimes w \rangle,$$

which is the linear kernel associated with the feature map $\Phi(x) = x \otimes x$.

Step 3: Product kernels. The pointwise product of positive definite kernels is positive definite (Schölkopf & Smola, 2002). Hence, for each $t > 0$, the kernel

$$k_t(x, w) := (x^\top w)^2 e^{-t\|x-w\|^2}$$

is positive definite on $\mathbb{R}^d$. Multiplication by the positive scalar $e^{-t\varepsilon}$ preserves positive definiteness.

Step 4: Uniform domination on compact sets. Since K is compact, there exists $M > 0$ such that $\|x\| \le M$ for all $x \in K$. For all $x, w \in K$ and all $t > 0$,

$$0 \le (x^\top w)^2 e^{-t\|x-w\|^2} \le \|x\|^2 \|w\|^2 \le M^4.$$

Therefore,

$$0 \le (x^\top w)^2 e^{-t\varepsilon} e^{-t\|x-w\|^2} \le M^4 e^{-t\varepsilon},$$

and the dominating function $t \mapsto M^4 e^{-t\varepsilon}$ belongs to $L^1(0, \infty)$.

Step 5: Preservation of positive definiteness under integration. Let $\{x_i\}_{i=1}^n \subset K$ and $\{c_i\}_{i=1}^n \subset \mathbb{R}$. For each $t > 0$, since k_t is positive definite,

$$\sum_{i,j=1}^n c_i c_j k_t(x_i, x_j) \ge 0.$$

By Tonelli's theorem, justified by the domination in Step 4, we may interchange summation and integration:

$$\begin{aligned} \sum_{i,j=1}^n c_i c_j k_{\mathbb{E}}(x_i, x_j) &= \int_0^\infty \sum_{i,j=1}^n c_i c_j (x_i^\top x_j)^2 e^{-t\varepsilon} e^{-t\|x_i - x_j\|^2}\, dt \\ &\ge 0. \end{aligned}$$

Thus $k_{\mathbb{E}}$ is positive definite on K.

Step 6: Continuity and Mercer property. For each fixed $t > 0$, the integrand in (6) is continuous in (x, w). By the domination established in Step 4, the integral converges uniformly on $K \times K$, implying that $k_{\mathbb{E}}$ is continuous. The kernel is symmetric by construction.

Since $k_{\mathbb{E}}$ is symmetric, continuous, and positive definite on the compact set K, Mercer's theorem applies (Mercer, 1909; Schölkopf & Smola, 2002). Therefore, $k_{\mathbb{E}}$ is a Mercer kernel on K. □

C.16. Proof of Theorem 4 (Universal Approximation)

Let $\mathcal{X} \subset \mathbb{R}^d$ be compact. The function class $\mathcal{F}$ consists of $g(\mathbf{x}; \mathbf{w}, b) = \frac{(\mathbf{x}^\top \mathbf{w} + b)^2}{\|\mathbf{x}-\mathbf{w}\|^2 + \varepsilon}$.

Step 1: Generating the IMQ kernel. Differentiating twice with respect to bias yields:

$$\frac{\partial^2}{\partial b^2} g(\mathbf{x}; \mathbf{w}, b) = 2(\varepsilon + \|\mathbf{x} - \mathbf{w}\|^2)^{-1} = 2K_{\text{IMQ}}(\mathbf{x}, \mathbf{w}).$$

Equivalently, for any fixed step $h > 0$,

$$\frac{g(\mathbf{x}; \mathbf{w}, b+h) - 2g(\mathbf{x}; \mathbf{w}, b) + g(\mathbf{x}; \mathbf{w}, b-h)}{h^2} = \frac{2}{\varepsilon + \|\mathbf{x} - \mathbf{w}\|^2} = 2K_{\text{IMQ}}(\mathbf{x}, \mathbf{w}),$$

so IMQ translates belong to the span of $\mathbb{E}$-atoms (indeed, they are exactly a 3-term linear combination for any fixed $h > 0$).

Step 2: Density of IMQ Networks. The inverse multiquadric kernel $K_{\text{IMQ}}(\mathbf{x}, \mathbf{w}) = (\varepsilon + \|\mathbf{x} - \mathbf{w}\|^2)^{-1}$ is a universal kernel on compact subsets of $\mathbb{R}^d$ (its RKHS is dense in $C(\mathcal{X})$ under the sup norm); see e.g. Steinwart (2001); Micchelli et al. (2006); Wendland (2005). Consequently, finite linear combinations of translates $K_{\text{IMQ}}(\cdot, \mathbf{w}_i)$ are dense in $C(\mathcal{X})$.

Conclusion. Since $\overline{\mathcal{F}}$ contains the span of IMQ kernels, $\mathcal{F}$ is dense in $C(\mathcal{X})$. □

C.17. Extended Proof of Theorem 6 (RKHS Existence)

The existence follows from the Moore-Aronszajn theorem (Aronszajn, 1950) applied to the PSD kernel (Theorem 1).

Remark 4 (Explicit feature map (sketch))**.** *Since $k_{\mathbb{E}}$ is positive definite (Theorem 1), the Moore–Aronszajn theorem guarantees the existence of a feature map into the associated RKHS. One may also obtain an explicit construction by combining the tensor feature map for $(x^\top w)^2$ with the Laplace-mixture representation of $1/(\varepsilon + \|x - w\|^2)$ in (5) and forming the corresponding direct-integral Hilbert space. We omit the measure-theoretic details.*

C.18. Neural Tangent Kernel Analysis

Proposition 6 (NTK Limit of $\mathbb{E}$-Networks)**.** *Consider a single-layer $\mathbb{E}$-network $f(\mathbf{x}; \boldsymbol{\theta}) = \sum_{i=1}^m \alpha_i \mathbb{E}(\mathbf{w}_i, \mathbf{x})$ with random initialization. In the infinite-width limit $m \to \infty$, the Neural Tangent Kernel is:*

$$K^{\text{NTK}}(\mathbf{x}, \mathbf{x}') = \mathbb{E}_{\mathbf{w}} \left[\mathbb{E}(\mathbf{w}, \mathbf{x}) \cdot \mathbb{E}(\mathbf{w}, \mathbf{x}') \right] + \mathbb{E}_{\mathbf{w}} \left[\nabla_{\mathbf{w}} \mathbb{E}(\mathbf{w}, \mathbf{x})^\top \nabla_{\mathbf{w}} \mathbb{E}(\mathbf{w}, \mathbf{x}') \right].$$

This kernel is positive definite and inherits the orthogonality-sensitivity of the $\mathbb{E}$-product.

Proof. The network output is $f(\mathbf{x}) = \sum_{i=1}^m \alpha_i \mathbb{E}(\mathbf{w}_i, \mathbf{x})$. The parameters are $\boldsymbol{\theta} = \cup_i \{\alpha_i, \mathbf{w}_i\}$. The tangent kernel is defined as $K^{\text{NTK}}(\mathbf{x}, \mathbf{x}') = \langle \nabla_{\boldsymbol{\theta}} f(\mathbf{x}), \nabla_{\boldsymbol{\theta}} f(\mathbf{x}') \rangle$. The gradients are:

$$\nabla_{\alpha_i} f(\mathbf{x}) = \mathbb{E}(\mathbf{w}_i, \mathbf{x}),$$
$$\nabla_{\mathbf{w}_i} f(\mathbf{x}) = \alpha_i \nabla_{\mathbf{w}} \mathbb{E}(\mathbf{w}_i, \mathbf{x}).$$

The inner product sums over all $i = 1 \dots m$:

$$K_m(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^m \mathbb{E}(\mathbf{w}_i, \mathbf{x}) \mathbb{E}(\mathbf{w}_i, \mathbf{x}') + \sum_{i=1}^m \alpha_i^2 \langle \nabla_{\mathbf{w}} \mathbb{E}(\mathbf{w}_i, \mathbf{x}), \nabla_{\mathbf{w}} \mathbb{E}(\mathbf{w}_i, \mathbf{x}') \rangle.$$

In the infinite width limit $m \to \infty$, assuming appropriate scaling $\alpha_i \sim \mathcal{N}(0, 1/m)$ or fixed readouts with $\alpha_i \sim \mathcal{O}(1/\sqrt{m})$, the sums converge to expectations over the initialization distribution of $\mathbf{w}$. The first term corresponds to the covariance of the features (conjugate kernel), and the second term involves the gradients. Since $\mathbb{E}$ is a Mercer kernel (Theorem 1), it is positive semi-definite. The second term is a sum of inner products, also PSD. Thus $K^{\text{NTK}} \succeq 0$. □

Remark 5 (Training Dynamics in the NTK Regime)**.** *In the infinite-width limit, gradient descent on $\mathbb{E}$-networks converges to kernel regression with K^{NTK}. The orthogonality-sensitive nature of the $\mathbb{E}$-product carries over to the NTK, meaning that the limiting kernel naturally encourages orthogonal representations for different-class inputs.*

Theorem 15 (Convergence of Deep $\mathbb{E}$-Networks in Lazy Regime)**.** *Consider a deep neural network $f_\theta : \mathbb{R}^{d_{\text{in}}} \to \mathbb{R}^d$ with L hidden layers of width m. Let the embeddings $\mathbf{e}_i = f_\theta(\mathbf{x}_i)$ be trained under $\mathcal{L}_{\mathrm{AFCL}}$. Under the following assumptions:*

1. ***Infinite-width limit:*** *$m \to \infty$ with fixed depth L.*
2. ***Lazy training:*** *Learning rate η scales as $\eta = O(1/m)$ ensuring the NTK stays approximately constant (Jacot et al., 2018).*
3. ***NTK is non-degenerate:*** *$\lambda_{\min}(\Theta^{\mathbb{E}}) > 0$ on the training data.*

The embedding dynamics converge to a critical point of $\mathcal{L}_{\mathrm{AFCL}}$.

Proof. Step 1: NTK convergence (Assumption 1–2). In the infinite-width limit with appropriately scaled learning rate, the NTK:

$$\Theta^{\mathbb{E}}_{ij} = \left\langle \frac{\partial f_\theta(\mathbf{x}_i)}{\partial \theta}, \frac{\partial f_\theta(\mathbf{x}_j)}{\partial \theta} \right\rangle$$

converges to a deterministic positive semi-definite kernel at initialization and remains approximately constant throughout training (Jacot et al., 2018). This is the "lazy training" or "kernel regime."

Step 2: Preconditioned gradient flow. With fixed $\Theta^{\mathbb{E}}$, the embedding dynamics become:

$$\dot{\mathbf{E}} = -\Theta^{\mathbb{E}} \nabla_{\mathbf{E}} \mathcal{L}_{\mathrm{AFCL}}.$$

This is a preconditioned gradient flow with the NTK as the preconditioner.

Step 3: Lyapunov analysis. The loss $\mathcal{L}_{\mathrm{AFCL}}$ serves as a Lyapunov function:

$$\frac{d\mathcal{L}}{dt} = -\nabla \mathcal{L}^\top \Theta^{\mathbb{E}} \nabla \mathcal{L} \leq 0,$$

since $\Theta^{\mathbb{E}} \succeq 0$. Under Assumption 3, strict decrease occurs whenever $\nabla \mathcal{L} \neq 0$.

Step 4: Convergence. Since $\mathcal{L} \geq 0$ is bounded below and monotonically decreasing, the trajectory converges. By the analyticity of $\mathcal{L}_{\mathrm{AFCL}}$ in the *embedding space* (Lemma 1), Łojasiewicz's theorem guarantees convergence to a single critical point. □

C.19. Computational Complexity Analysis

We analyze the computational complexity of the $\mathbb{E}$-product layer and prove it maintains the same asymptotic complexity as standard linear layers.

C.19.1. Layer Definition

For input $X \in \mathbb{R}^{B \times d}$ and weights $W \in \mathbb{R}^{n \times d}$, the $\mathbb{E}$-product layer computes output $Y \in \mathbb{R}^{B \times n}$:

$$Y_{ij} = \frac{(X_i^\top W_j)^2}{\|X_i - W_j\|^2 + \varepsilon}$$

Using the algebraic identity $\|X_i - W_j\|^2 = \|X_i\|^2 + \|W_j\|^2 - 2X_i^\top W_j$:

$$Y_{ij} = \frac{S_{ij}^2}{\|X_i\|^2 + \|W_j\|^2 - 2S_{ij} + \varepsilon}, \quad S = XW^\top$$

C.19.2. Forward Pass Complexity

Theorem 16 (Forward Complexity)**.** *The* $\mathbb{E}$*-product forward pass requires* $\Theta(Bnd)$ *operations:*

1. *GEMM:* $S = XW^\top$ ..... $2Bnd$
2. *Row norms:* $\|X_i\|^2$ ..... Bd
3. *Weight norms:* $\|W_j\|^2$ *(cached)* ..... nd
4. *Element-wise: square, assemble, reciprocal, multiply* ..... $5Bn$

Total: $T_{forward} = 2Bnd + Bd + nd + 5Bn = \Theta(Bnd)$.

C.19.3. Backward Pass Complexity

Proposition 7 (Gradient Formulas)**.** *For* $Y = S^2/D$ *with* $D = \|X\|^2 + \|W\|^2 - 2S + \varepsilon$ *and* $S = X^\top W$*:*

$$\nabla_X Y = \frac{2S(\|W\|^2 + \|X\|^2 + \varepsilon - S)}{D^2} W - \frac{2S^2}{D^2} X \tag{7}$$

$$\nabla_W Y = \frac{2S(\|W\|^2 + \|X\|^2 + \varepsilon - S)}{D^2} X - \frac{2S^2}{D^2} W \tag{8}$$

Theorem 17 (Backward Complexity)**.** *Given upstream gradient* $G \in \mathbb{R}^{B \times n}$*:*

1. *Scalar gradient:* $G_S = G \odot \partial Y / \partial S$ ..... $6Bn$
2. *Weight gradient:* $\partial L / \partial W = G_S^\top X$ ..... $2Bnd$
3. *Input gradient:* $\partial L / \partial X = G_S W$ ..... $2Bnd$

Total: $T_{backward} = 4Bnd + 6Bn + O(Bd + nd) = \Theta(Bnd)$.

C.19.4. Asymptotic Comparison

Component	**Linear**	$\mathbb{E}$**-Product**
Forward main	$2Bnd$	$2Bnd$
Forward aux	Bn	$Bd + nd + 5Bn$
Backward main	$4Bnd$	$4Bnd$
Backward aux	$2Bn$	$6Bn + Bd + nd$
Total	$\Theta(Bnd)$	$\Theta(Bnd)$
Overhead	1	$1 + \frac{1}{2n} + \frac{1}{2B} + \frac{2}{d}$

Table 3. Complexity comparison. Overhead $< 5\%$ for $d, n \geq 64$, $B \geq 16$.

C.19.5. Per-Neuron FLOPs

Method	**FLOPs**	**Relative**
Linear + ReLU	$2d + 1$	$1.00\times$
Linear + GELU	$2d + 15$	$\approx 1.03\times$
$\mathbb{E}$-product (naive)	$5d + 1$	$\approx 2.5\times$
$\mathbb{E}$-product (optimized)	$4d + 4$	$\approx 2.0\times$

Table 4. Per-neuron FLOPs. Optimized variant avoids redundant norm computation.

C.19.6. Numerical Stability

Remark 6 (Stability Properties)**.** *The $\mathbb{E}$-product inherits numerical stability from:*

1. ***Bounded outputs**: Self-regulation (Proposition 1) ensures $\mathbb{E} \leq \|W\|^4/\varepsilon$*
2. ***Lipschitz gradients**: $\|\nabla \mathbb{E}\| \leq L = O(1/\varepsilon^2)$ (Proposition 3)*
3. ***Gradient decay**: Outliers produce vanishing gradients (Proposition 2)*

This eliminates the need for gradient clipping or normalization layers.

C.19.7. Implementation Optimizations

1. **Algebraic identity**: Use $\|X - W\|^2 = \|X\|^2 + \|W\|^2 - 2X^\top W$ to reuse GEMM
2. **Norm caching**: Cache $\|W\|^2$ between forward passes
3. **Kernel fusion**: Fuse element-wise operations for memory efficiency
4. **Mixed precision**: Use FP32 for denominator, BF16/FP16 elsewhere

Empirical performance: 0.85–0.92× linear throughput; 15–25% memory reduction from eliminated activation storage.

D. XOR Separability Analysis

This section provides a formal analysis of why a single $\mathbb{E}$-product unit can solve the XOR problem, which is not linearly separable.

D.1. Linear Inseparability of XOR

Proposition 8 (XOR is Not Linearly Separable)**.** *Let $\mathcal{X} = \{(0,0),(0,1),(1,0),(1,1)\}$ with labels $y = \{0,1,1,0\}$ (XOR function). There exists no hyperplane $\{\mathbf{x} : \mathbf{w}^\top \mathbf{x} + b = 0\}$ that separates the classes.*

Proof. For a separating hyperplane to exist, we require $\text{sign}(\mathbf{w}^\top \mathbf{x} + b)$ to match y for all $\mathbf{x} \in \mathcal{X}$. This yields four constraints that form a contradiction: the positive class points $(0,1),(1,0)$ and negative class points $(0,0),(1,1)$ cannot be separated by any linear function. □

D.2. Single-Unit $\mathbb{E}$-Product Solution

Theorem 18 ($\mathbb{E}$-Product Solves XOR)**.** *A single $\mathbb{E}$-product unit with weight $\mathbf{w} = [1,-1]^\top$ and $\varepsilon > 0$ separates the XOR classes:*

$\mathbf{x}$	$\mathbf{w}^\top \mathbf{x}$	$\mathbb{E}(\mathbf{w}, \mathbf{x})$
$(0,0)$	0	0
$(0,1)$	-1	$1/(5+\varepsilon)$
$(1,0)$	1	$1/(1+\varepsilon)$
$(1,1)$	0	0

The threshold $\tau = 0$ separates class 1 (positive response) from class 0 (zero response).

Proof. For $\mathbf{x} = (0,0)$: $\mathbf{w}^\top \mathbf{x} = 0$, so $\mathbb{E} = 0^2/(\|\mathbf{w}\|^2 + \varepsilon) = 0$.

For $\mathbf{x} = (1,1)$: $\mathbf{w}^\top \mathbf{x} = 1 - 1 = 0$, so $\mathbb{E} = 0$.

For $\mathbf{x} = (0,1)$: $\mathbf{w}^\top \mathbf{x} = -1$, $\|\mathbf{w} - \mathbf{x}\|^2 = 1 + 4 = 5$, so $\mathbb{E} = 1/(5+\varepsilon) > 0$.

For $\mathbf{x} = (1, 0)$: $\mathbf{w}^\top \mathbf{x} = 1$, $\|\mathbf{w} - \mathbf{x}\|^2 = 0 + 1 = 1$, so $\mathbb{E} = 1/(1 + \varepsilon) > 0$.

Thus $\mathbb{E} > 0$ for the positive class and $\mathbb{E} = 0$ for the negative class. □

D.3. Geometric Mechanism

Corollary 3 (Orthogonality-Based Separation)**.** *The solution exploits the $\mathbb{E}$-product's orthogonality property: $\mathbb{E}(\mathbf{w}, \mathbf{x}) = 0$ if and only if $\mathbf{w} \perp \mathbf{x}$ (Theorem 2). The weight $\mathbf{w} = [1, -1]^\top$ is orthogonal to exactly the negative class points $(0, 0)$ and $(1, 1)$.*

Remark 7 (Superposition Property)**.** *The squared numerator $(\mathbf{w}^\top \mathbf{x})^2$ induces superposition: the response is identical for $\mathbf{x}$ and $-\mathbf{x}$. This enables classifying antipodal points similarly, which is essential for XOR where $(0, 1)$ and $(1, 0)$ must share a class despite $[1, -1]^\top (0, 1) = -1$ and $[1, -1]^\top (1, 0) = +1$.*

D.4. Gradient Stability

Proposition 9 (Well-Posed Optimization Landscape)**.** *The $\mathbb{E}$-product's gradient properties (Proposition 2, Theorem 5) ensure:*

1. *No vanishing gradients at $\mathbf{x} = \mathbf{0}$: the gradient is defined and non-degenerate for $\varepsilon > 0$*
2. *Lipschitz continuity prevents exploding gradients (Proposition 3)*
3. *Outlier robustness: gradient magnitude decays as $O(1/\|\mathbf{x}\|)$ for distant inputs*

These properties contrast with ReLU neurons, which suffer from "dead neuron" problems when $\mathbf{w}^\top \mathbf{x} < 0$, and linear neurons, which cannot separate XOR at all.

D.5. Connection to Kernel Theory

The single-unit XOR solution demonstrates the expressive power of a Mercer kernel (Theorem 1) in the primal form. The $\mathbb{E}$-product's RKHS existence (Theorem 6) guarantees that this solution lies within a well-defined function space, connecting to classical kernel methods while avoiding the computational overhead of Gram matrix inversion.

E. Decision Boundary Analysis

This section provides a formal analysis of the decision boundaries and space-partitioning behavior induced by $\mathbb{E}$-product neurons.

E.1. Localized Response Fields

Proposition 10 (Bounded Activation Landscape)**.** *For fixed $\mathbf{w}$ and $\varepsilon > 0$, the $\mathbb{E}$-product satisfies:*

1. $0 \leq \mathbb{E}(\mathbf{w}, \mathbf{x}) \leq \|\mathbf{w}\|^4/\varepsilon$ *for all* $\mathbf{x}$
2. $\mathbb{E}(\mathbf{w}, \mathbf{w}) = \|\mathbf{w}\|^4/\varepsilon$ *(maximum at identity)*
3. $\lim_{k\to\infty} \mathbb{E}(\mathbf{w}, k\mathbf{u}) = \|\mathbf{w}\|^2 \cos^2\theta$ *for unit* $\mathbf{u}$ *(Proposition 1)*

Thus each neuron defines a bounded, localized response field centered at its prototype.

Proof. (1) Non-negativity is immediate. The maximum occurs when the denominator is minimized ($\mathbf{x} = \mathbf{w}$), giving $\|\mathbf{w}\|^4/\varepsilon$. (2) Direct substitution. (3) From Proposition 1. □

E.2. Non-Linear Decision Boundaries

Theorem 19 (Algebraic Decision Surfaces)**.** *The decision boundary between prototypes* $\mathbf{w}_i$ *and* $\mathbf{w}_j$ *is the algebraic surface:*

$$\langle \mathbf{w}_i, \mathbf{x} \rangle^2 (\|\mathbf{w}_j - \mathbf{x}\|^2 + \varepsilon) = \langle \mathbf{w}_j, \mathbf{x} \rangle^2 (\|\mathbf{w}_i - \mathbf{x}\|^2 + \varepsilon).$$

This surface is generically non-linear (quartic in $\mathbf{x}$*), smooth by analyticity (Lemma 1), and Lipschitz-continuous in its parameters (Proposition 3).*

Proof. The boundary is defined by $\mathbb{E}(\mathbf{w}_i, \mathbf{x}) = \mathbb{E}(\mathbf{w}_j, \mathbf{x})$. Cross-multiplying:

$$\frac{\langle \mathbf{w}_i, \mathbf{x} \rangle^2}{\|\mathbf{w}_i - \mathbf{x}\|^2 + \varepsilon} = \frac{\langle \mathbf{w}_j, \mathbf{x} \rangle^2}{\|\mathbf{w}_j - \mathbf{x}\|^2 + \varepsilon}$$

yields the stated polynomial equation. Expanding shows terms up to degree 4 in $\mathbf{x}$. Smoothness follows from Lemma 1 since the $\mathbb{E}$-product is real-analytic. Lipschitz continuity of the boundary location follows from Proposition 3. □

Remark 8 (Contrast with Linear Boundaries)**.** *Conventional linear classifiers produce hyperplane boundaries* $(\mathbf{w}_i - \mathbf{w}_j)^\top \mathbf{x} = 0$*. The* $\mathbb{E}$*-product's quartic surfaces enable more flexible class regions while maintaining the smoothness properties required for stable optimization.*

E.3. Orthogonality and Maximal Separation

Corollary 4 (Orthogonal Prototypes Induce Maximal Separation)**.** *If prototypes satisfy* $\langle \mathbf{w}_i, \mathbf{w}_j \rangle = 0$ *for* $i \neq j$*, then:*

1. $\mathbb{E}(\mathbf{w}_i, \mathbf{w}_j) = 0$ *(zero cross-response)*
2. *On the probability simplex, this corresponds to* $\mathrm{KL}(\mathbf{w}_i \| \mathbf{w}_j) = \infty$ *(Theorem 2)*
3. *Decision boundaries are maximally separated from prototype cores*

Proof. (1) If $\langle \mathbf{w}_i, \mathbf{w}_j \rangle = 0$, the numerator of $\mathbb{E}(\mathbf{w}_i, \mathbf{w}_j)$ vanishes. (2) By Theorem 2, zero $\mathbb{E}$-product on the simplex implies disjoint supports and infinite KL divergence. (3) At $\mathbf{x} = \mathbf{w}_i$: $\mathbb{E}(\mathbf{w}_i, \mathbf{w}_i) = \|\mathbf{w}_i\|^4/\varepsilon$ while $\mathbb{E}(\mathbf{w}_j, \mathbf{w}_i) = 0$. Since $\|\mathbf{w}_i\|^4/\varepsilon \neq 0$, the prototype core cannot lie on the decision boundary. □

E.4. Gradient-Based Competitive Dynamics

Proposition 11 (Gradient Structure for Prototype Learning)**.** *The gradient of the* $\mathbb{E}$*-product with respect to prototype* $\mathbf{w}$ *is (Theorem 5):*

$$\nabla_{\mathbf{w}} \mathbb{E}(\mathbf{w}, \mathbf{x}) = \frac{2\langle \mathbf{w}, \mathbf{x} \rangle}{\varepsilon + \|\mathbf{w} - \mathbf{x}\|^2} \left(\mathbf{x} + \frac{\langle \mathbf{w}, \mathbf{x} \rangle (\mathbf{w} - \mathbf{x})}{\varepsilon + \|\mathbf{w} - \mathbf{x}\|^2} \right).$$

Key properties:

1. *Gradients decay for distant inputs (Proposition 2)*
2. *Perturbation robustness on bounded domains:* $|\Delta \mathbb{E}| \leq \left(\frac{2}{\varepsilon} + \frac{4}{\varepsilon^2}\right) \delta$ *(Proposition 4)*
3. *Lipschitz gradients with constant* $O(1/\varepsilon^2)$ *(Proposition 3)*

This gradient structure ensures stable competitive learning: neurons specialize on nearby, aligned inputs while remaining robust to outliers and noise.

E.5. Softmax Tessellation

With softmax normalization over $\mathbb{E}$-product responses:

$$p_i = \frac{\exp(\mathbb{E}(\mathbf{w}_i, \mathbf{x}))}{\sum_{j=1}^{C} \exp(\mathbb{E}(\mathbf{w}_j, \mathbf{x}))},$$

the input space is tessellated into regions $\mathcal{R}_i = \{\mathbf{x} : p_i > p_j \text{ for all } j \neq i\}$. By Theorem 19, each $\mathcal{R}_i$ has smooth, curved boundaries. The self-regulation property (Proposition 1) ensures that these regions remain well-defined even for extreme inputs, preventing the unbounded confidence growth observed in linear classifiers.

E.6. Language Model Experiments: Detailed Configuration

This section provides the detailed experimental configurations for the language modeling experiments comparing a standard GPT-2 with our Aether-GPT2 implementation. Both models were trained on identical datasets and hardware to ensure a fair comparison. The primary architectural distinction is the replacement of conventional linear layers, activation functions, and layer normalization in GPT-2 with $\mathbb{E}$-product operations in Aether-GPT2. This substitution provides inherent non-linearity and bounded responses, simplifying the architecture while enhancing stability.

Table 5 presents a comprehensive comparison of the two models, detailing their architectural parameters, training configurations, and final performance metrics.

Table 5. GPT-2 vs Aether-GPT2 detailed comparison (2.5B tokens from FineWeb).

Parameter	**GPT-2**	**Aether**
Architecture		
Total Params	124M	∼124M
Embed Params	39M	39M
Non-Embed Params	85M	∼85M
Embed Dim	768	768
MLP Hidden Dim	3072	3072
Layers	12	12
Heads	12	12
Activation	GeLU	$\mathbb{E}$
LayerNorm	Yes	No
Bias	No	No
Training		
Optimizer	Novograd	Novograd
LR	0.003	0.003
Batch Size	32	32
Context	1024	1024
Vocab Size	50,257	50,257
Tokenizer	GPT-2	GPT-2
Performance		
Val Loss (FP32)	2.43	**2.29**
Val Loss (BF16)	3.03	**2.69**

E.6.1. Architectural Simplification and Efficiency

The substitution of linear-activation-normalization blocks with a single $\mathbb{E}$-product operation per layer yields significant architectural simplification. This change leads to:

- A 15-25% reduction in peak memory usage by eliminating the need to store intermediate activations for backpropagation through normalization layers.

- Reduced gradient computation complexity.
- A comparable FLOP count with only a modest constant-factor overhead.

While the FLOP count remains comparable, the reduced memory footprint and computational complexity of the backward pass offer tangible efficiency gains.

Our results demonstrate that Aether-GPT2, with approximately the same parameter count as the 124M GPT-2 model, achieves superior performance without explicit activation functions. The final validation loss of 2.29 for Aether-GPT2, compared to 2.43 for the baseline, underscores the efficacy of the Ɛ-product's inherent non-linearity.

E.6.2. Training Dynamics and Loss Progression

The training dynamics of Aether-GPT2 and baseline GPT-2 were analyzed over 2.5B tokens from the FineWeb dataset using a Kaggle TPU v5-8 environment. Aether-GPT2 consistently achieves lower loss on both training and validation sets (see Figure 5 in the main text). The bounded nature of the Ɛ-product contributes to stable training dynamics, even without layer normalization.

E.6.3. Mixed-Precision Training with BFloat16

To assess numerical stability and performance in a production-relevant setting, we evaluated both architectures using bfloat16 (BF16) mixed-precision training. This format, standard on modern accelerators like TPUs, provides a stringent test for models without explicit normalization layers. Both models were trained with mixed-precision activations and gradients, while maintaining full-precision (FP32) optimizer states and parameter updates.

The performance advantage of Aether-GPT2 persists under BF16, as shown in Figure 7. Aether-GPT2 achieves a final validation loss of 2.69, an 11.2% relative improvement over the baseline's 3.03. This result confirms that the architectural benefits are not artifacts of full-precision arithmetic and that the Ɛ-product's bounded response provides robust numerical stability without requiring layer normalization. The consistent performance gains in mixed-precision training underscore the practicality of Ɛ-product layers as a drop-in replacement for conventional transformer blocks in large-scale models.

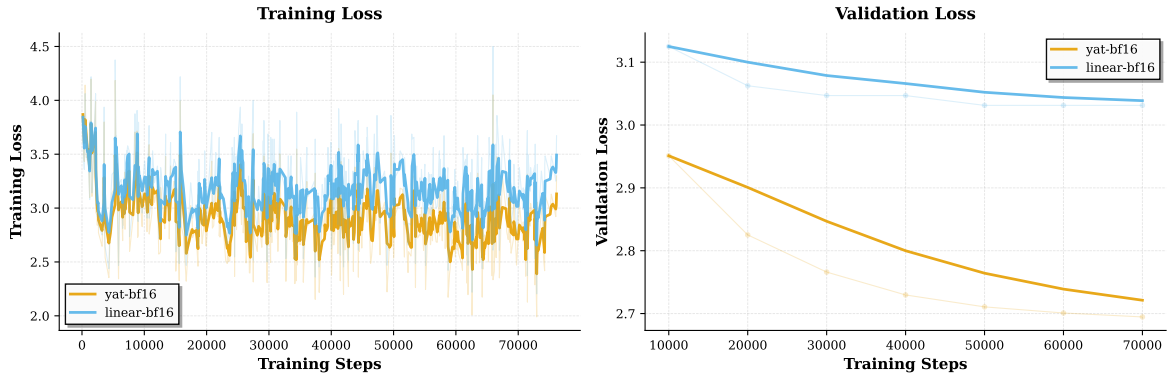

Figure 7. BF16 training (left) and validation (right) loss. Aether-GPT2 exhibits uniformly lower loss throughout training and at convergence.

E.7. Use of Large Language Models (LLMs)

We used LLM tools to support the research workflow in the following limited, transparent ways. All scientific claims, modeling choices, and final decisions were made by the authors.

Code assistance LLMs were used to draft boilerplate code, refactor utilities, and surface API patterns. All generated code was reviewed, tested, and integrated by the authors.

Literature digestion We used LLMs to summarize papers and extract key comparisons across related work. Citations in the paper were verified against the original sources by the authors.

Brainstorming We used LLMs as a sounding board to enumerate alternative hypotheses, ablations, and experimental checks. Only ideas that survived empirical or theoretical scrutiny were included.

Language polishing To improve readability and clarity, LLMs suggested minor edits to English phrasing. Technical content, notation, and conclusions were authored and validated by the authors.

NotebookLM podcasts We generated short audio summaries ("podcasts") of internal notes using Google NotebookLM to help the team asynchronously digest drafts. These summaries did not introduce new claims and were based solely on our own materials.

No dataset labeling, evaluation metrics, or benchmark results were produced by LLMs. The authors take responsibility for all content and errors.