Fast Neural Kernel Embeddings for General Activations

Insu Han[1] **Amir Zandieh**[2] **Jaehoon Lee**[3]
Roman Novak[3] **Lechao Xiao**[3] **Amin Karbasi**[1,3]

[1]Yale University [2]Max-Planck-Institut für Informatik [3]Google Research

Abstract

Infinite width limit has shed light on generalization and optimization aspects of deep learning by establishing connections between neural networks and kernel methods. Despite their importance, the utility of these kernel methods was limited in large-scale learning settings due to their (super-)quadratic runtime and memory complexities. Moreover, most prior works on neural kernels have focused on the ReLU activation , mainly due to its popularity but also due to the difficulty of computing such kernels for general activations. In this work, we overcome such difficulties by providing methods to work with general activations. First, we compile and expand the list of activation functions admitting exact dual activation expressions to compute neural kernels. When the exact computation is unknown, we present methods to effectively approximate them. We propose a fast sketching method that approximates any multi-layered Neural Network Gaussian Process (NNGP) kernel and Neural Tangent Kernel (NTK) matrices for a wide range of activation functions, going beyond the commonly analyzed ReLU activation. This is done by showing how to approximate the neural kernels using the truncated Hermite expansion of any desired activation functions. While most prior works require data points on the unit sphere, our methods do not suffer from such limitations and are applicable to any dataset of points in $\mathbb{R}^d$. Furthermore, we provide a subspace embedding for NNGP and NTK matrices with near input-sparsity runtime and near-optimal target dimension which applies to any *homogeneous* dual activation functions with rapidly convergent Taylor expansion. Empirically, with respect to exact convolutional NTK (CNTK) computation, our method achieves $106\times$ speedup for approximate CNTK of a 5-layer Myrtle network on CIFAR-10 dataset.

1 Introduction

Infinite width limit has enabled fundamental understandings of deep neural networks by establishing a correspondence to kernel methods. In this limit, the network's function prior is a Gaussian process [1–3] and under gradient descent training with squared loss, the network behaves as a linearized function [4, 5]. Underlying these limit, a core object is a neural kernel which encapsulates architectural inductive prior in its functional form [6]. The kernel describing gradient descent dynamics, the Neural Tangent Kernel (NTK) [4], and Neural Network Gaussian Process (NNGP) [2] kernel have been extensively studied [7–12] since they were initially identified. In particular, the infinite width theory has shed light on powerful abilities of deep neural networks including optimization [13–16], generalization [17–19], regularization [20–22] and robustness [23, 24]. Beyond theoretical findings, it has been extensively reported that neural kernels can enhance practical applications including small data classification/regression tasks [25], neural architect search [26, 27], dataset distillation [28, 29], federated learning [30], meta learning [31], generalization attack [32], just to name a few.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

Despite those powerful advantages, there is still a gap between practice and theory in the utility of these kernel methods. First, the NNGP and NTK can be exactly computed recursively [2, 4] however, the explicit forms are only known when the corresponding neural networks contain a few set of activation functions such as ReLU or Error functions. While ReLU activation is the default choice for many deep learning applications, recently different activation functions have shown to work well in various domains of machine learning. For example, GeLU [33] has been widely used in Transformer based natural language processing settings [34–36] and sinusoidal activation functions work well for implicit neural representation (e.g. NeRF) [37, 38]. Moreover, Xie et al. [39] showed that smooth activation functions could improve robustness compared to ReLU-based models. To enable better theoretical understanding on the role of these activation functions in these domain, expanding the infinite width limit tool set to general activation function is an important step forward.

Secondly, even if the exact neural kernel computation is explicitly known, it requires significantly huge amount of computing resources. For example, it will take order of few 100 to 1,000 GPU hours to compute the exact NTK of depth 10 convolutional neural networks with pooling on 60,000 CIFAR-10 dataset. High compute requirement is often too expensive to perform extensive studies or use in a practical setting. While Novak et al. [40] have sped up Monte Carlo estimation of the NTK, random sampling remains impractical due to still high kernel computation cost, and cubic (in the training set size) inference cost. Recently, Zandieh et al. [41] proposed an efficient method to approximate the NTK computation via sketching algorithms. Their algorithm can approximate the neural kernels with ReLU activation orders of magnitude faster than the exact one. But it remains unclear how sketching algorithms are extended to other activations.

In this work, we fill this gap by showing that neural kernel for *arbitrary* smooth activation can be expressed in a form of series expansion. We first focus on how to express a kernel function of neural network with a single hidden layer. Under the infinite width limit, this kernel converges to a static function, so-called a *dual kernel*, and is determined by activation in the network. This is a key block to compute the NNGP and NTK of deeper architectures. We establish an explicit expression of dual kernel by expanding activation with the Hermite polynomial basis, and combining it with the fact that Hermite polynomials can play a role of random features of monomial kernels. As a result, our dual kernel formulation relies on coefficients of series expansion of the activation. In addition, we also derive dual kernel expression of the first-order derivative of activation. The NTK can be computed by combining these kernel computations. To the best of our knowledge, our work is the first to study the computation of the NTK for general activations. Furthermore, we provide a subspace embedding for NNGP and NTK matrices with near input-sparsity runtime and near-optimal target dimension. As activation functions play an important role in modern neural network architectures, we hope our work could empower researchers to explore properties of activations in a more principled way. Our main contributions are summarized as follows:

- **Building blocks for infinite-width neural kernel computations**: We derive an explicit expression of the dual kernel for a polynomial activation, which can be a building block for infinite-width neural kernel computations. For non-polynomial activation, we suggest to use its truncated Hermite expansion and analyze an error bound of the dual kernel.

- **Compiling and expanding dual activation Table 1**: We compile various known dual kernel for point-wise activations providing pointers to the original work and expand the set further. We hope our work also serve as an easy reference for various analytic expressions. We emphasize that while many prior references lack required computation for NTK, this work is comprehensive in covering both NNGP/NTK transformations for various activations where analytic computation is possible.

- **NTK computation**: Dual kernels of both activation and its derivative are essential for the NTK computation. Since our formulation requires coefficients of Taylor series of the activation, it is applicable to the dual kernel of derivative of the activation. In addition, we propose how to automatically compute the dual kernel of the derivative without knowing the activation. This approach is useful to characterize the NTK for kernel functions whose activation function is unavailable, e.g., normalized Gaussian, or whose dual kernel of the derivative is unavailable, e.g., GeLU and ELU.

- **Kernel approximation**: We analyze a pointwise error bound of approximated dual kernel via truncated Hermite expansion of the activation with a finite degree. The estimation error can decay polynomially faster in the degree. Furthermore, due to specific decomposition of our kernel formulation, we accelerate the NTK approximation by sketching techniques, similar to [41]. We also propose a new sketching method for the Convolutional NTK with homogeneous activations

Table 1: Activation functions and references for their dual kernels. More detailed expressions are provided in Appendix F.

Activation	$\sigma(t)$	Reference for the NNGP	Reference for the NTK
Rectified monomials	$t^q \cdot \mathbb{1}_{\{t \geq 0\}}$	[44]	[44]
Error function	$\mathrm{erf}(t)$	[43]	[5]
ABReLU (Leaky ReLU)	$-A \min(t, 0) + B \max(t, 0)$	[42, 50, 51]	[42, 50, 51]
Exponential	$\exp(At)$	[46, 52]	[46, 52]
Hermite polynomials	$h_q(t)$	[46]	This work
Sinusoidal	$\sin(At + B)$	[45, 47, 53]	This work
Gaussian	$\exp\left(-At^2\right)$	[43]	This work
GeLU	$\frac{t}{2}\left(1 + \mathrm{erf}\left(\frac{t}{\sqrt{2}}\right)\right)$	[48]	This work
ELU	$\mathrm{step}(t)t + \mathrm{step}(-t)\left(e^t - 1\right)$	[48]	This work
Normalized Gaussian	Unknown	[54]	This work
RBF	$\sqrt{2}\sin(\sqrt{2A}t + \frac{\pi}{4})$	[45]	This work
Gabor	$\exp(-t^2)\sin(t)$	This work	This work
Monomial	t^q	This work	This work
Polynomial	$\sum_{j=0}^{q} a_j t^j$	This work	This work

and analyze both a pointwise error bound and its runtime in Appendix D.2. Notably, our sketching method's runtime scales only linearly in the number of pixels of the input images, while the exact CNTK computation scales quadratically in the number of pixels.

- **Implementation**: We open-source NNGP and NTK for new activations within the Neural Tangents library [42] and sketching algorithm at `https://github.com/insuhan/ntk_activations`.

1.1 Related Work

Neural kernels (NTK, NNGP) can be computed using the recursive formula [2–5]. A prerequisite for these kernels is computing a static kernel function which is defined as the expectation of some function of (non-linear) activation in neural network over the standard normal distribution. Williams [43] studied this a dual kernel of $\mathrm{erf}(t)$ and Gaussian. Cho and Saul [44] derived dual kernels for the rectified monomials, i.e., $t^q \mathbb{1}_{\{t \geq 0\}}$, this function is equal to arc-cosine kernels where ReLU activation is a special case when $q = 1$. Rahimi and Recht [45] showed that sinusoidal activations, e.g., $\sin$ or $\cos$, can result in the Gaussian RBF kernel function using the Fourier transform. Daniely et al. [46] proposed a method to obtain a dual kernel if activation can be expanded by Hermite polynomials. However, inputs of the resulting kernels are restricted to be on the unit sphere. Louart et al. [47] analyzed asymptotic properties of dual kernel with random matrix theory and show closed-form formula of such as erf, $|t|$, sinusoidal. Tsuchida et al. [48] studied the dual kernels of both Gaussian Error Linear Unit (GeLU) [33] and Exponential Linear Unit (ELU) [33]. For activation that does not admit a closed-form expression, Lee et al. [2] numerically computed dual activation by doing interpolation on predetermined grid of variances and covariances. Table 1 summarizes activations whose dual kernels were priorly known, as well as expanding (in this work) the set to previously unknown expressions. Recently, Simon et al. [49] discovered that NTK of fully-connected neural network with any depth can be converted into that of a 1 hidden-layer neural network by modifying activation function. However, their method is limited to the normalized input data and fully-connected networks.

2 Preliminaries

Notations. We denote the identity matrix of dimension d by $\boldsymbol{I}_d$. For a scalar function f, we write $f^{(k)}$ to denote its k-th derivative. We use $\mathbb{1}_{\mathcal{E}}$ to denote the indicator of event $\mathcal{E}$. For a smooth function $\sigma : \mathbb{R} \to \mathbb{R}$, we use $\sigma^{(k)}$ to denote its k-th derivative and define $\|\sigma\|_{\mathcal{N}(0,\nu^2)}^2 := \mathbb{E}_{t \sim \mathcal{N}(0,\nu^2)}[|\sigma(t)|^2]$ for some $\nu \in \mathbb{R}$ and simply write $\|\sigma\|_{\mathcal{N}(0,1)} := \|\sigma\|_{\mathcal{N}}$. For scalar functions f, g we use $f \circ g$ to denote the composition of these functions and $f^{\circ q}$ to denote the q times self-composition of f, e.g., $f^{\circ 3}(x) = f(f(f(x)))$. Given a positive semidefinite matrix $\boldsymbol{K}$ and $\lambda > 0$, the statistical dimension of $\boldsymbol{K}$ with regularizer λ is defined as $s_\lambda(\boldsymbol{K}) := \mathtt{tr}(\boldsymbol{K}(\boldsymbol{K} + \lambda \boldsymbol{I})^{-1})$. We use $\mathrm{nnz}(x)$ to denote the number of nonzero entries in x. Given $x \in \mathbb{R}^m$ and $y \in \mathbb{R}^n$, we define $x \otimes y :=$

$[x_1y_1, x_2y_1, \dots x_my_1, x_1y_2, \dots x_my_2, \dots x_my_n]$ and $x^{\otimes p}$ as the p-fold self-tensoring of x. We also define $\oplus$ as the direct sum between vectors.

Hermite polynomials. The *Probabilist's Hermite polynomials* of degree $\ell \geq 0$ is defined as

$$h_\ell(t) = (-1)^\ell e^{\frac{t^2}{2}} \left[\frac{d^\ell}{dt^\ell} e^{-\frac{t^2}{2}}\right] = \ell! \sum_{i=0}^{\lfloor \ell/2 \rfloor} \frac{(-1)^i}{i!(\ell-2i)!} \frac{t^{\ell-2i}}{2^i}. \tag{1}$$

The polynomials $\{h_\ell\}_{\ell\geq 0}$ form a set of orthogonal basis for the space of square-integrable functions in $\mathbb{R}$ with respect to the normal measure $\mathcal{N}(0,1)$, i.e., the L^2 space of functions $L^2(\mathbb{R}, \mathcal{N}) := \{f : \mathbb{R} \to \mathbb{R} \mid \|\sigma\|_{\mathcal{N}}^2 < \infty\}$. Particularly, it holds that $\mathbb{E}_{t\sim\mathcal{N}(0,1)}\left[h_\ell(t)\, h_m(t)\right] = \ell! \cdot \mathbb{1}_{\{\ell=m\}}$. Thus, any function $f \in L^2(\mathbb{R}, \mathcal{N})$ has a unique Hermite expansion in the sense of $\|f - \sum_{t=0}^{\infty} c_j h_j\|_{\mathcal{N}} = 0$ and coefficient c_j can be computed as $c_j = \mathbb{E}_{t\sim\mathcal{N}(0,1)}\left[f(t)\, h_j(t)\right]/j!$.

Infinite width neural kernels. Given an activation $\sigma : \mathbb{R} \to \mathbb{R}$ satisfying that $\|\sigma\|_{\mathcal{N}} = 1$, consider a fully-connected L-layered neural network $f : \mathbb{R}^d \to \mathbb{R}$ for $L \geq 2$ defined as[1]

$$f_\sigma(x; \mathcal{W}) = \left\langle w^{(L)}, z_{L-1} \right\rangle / \sqrt{d_{L-1}}, \;\; z_\ell = \sigma\left(\boldsymbol{W}^{(\ell)} z_{\ell-1} / \sqrt{d_{l-1}}\right), \;\; z_0 = x \tag{2}$$

where $\mathcal{W} := \mathrm{vec}\left(w^{(L)}, \cup_{\ell=1}^{L-1} \boldsymbol{W}^{(\ell)}\right)$ for $w^{(L)} \in \mathbb{R}^{d_{L-1}}, \boldsymbol{W}^{(\ell)} \in \mathbb{R}^{d_\ell \times d_{\ell-1}}, d_0 := d, d_l := m$ for $l > 0$ is a collection of learnable parameters, m is the width of the network, and $\sigma(\cdot)$ is applied point-wisely. In the infinite width limit, i.e., $m \to \infty$, when all elements of $\mathcal{W}$ are initialized by i.i.d. random samples from $\mathcal{N}(0,1)$ and optimized via gradient descent on the least-square loss with an infinitesimal learning rate, the prediction of trained network becomes identical to that of its first order Taylor approximation at $\mathcal{W}$. Hence, inference with such ultra-wide network is equivalent to kernel regression with a static kernel, the so-called Neural Tangent Kernel (NTK), defined as $\Theta_\sigma^{(L)}(x,y) := \mathrm{plim}_{m\to\infty} \langle \nabla_{\mathcal{W}} f_\sigma(x;\mathcal{W}), \nabla_{\mathcal{W}} f_\sigma(y;\mathcal{W})\rangle$ (convergence in probability to a constant). In addition, at initialization the output of an infinitely wide network is equivalent to a sample from a Gaussian process with mean zero and covariance $\Sigma_\sigma^{(L)}(x,y) := \mathrm{plim}_{m\to\infty} \langle f_\sigma(x;\mathcal{W}), f_\sigma(y;\mathcal{W})\rangle$, known as the Neural Network Gaussian Process (NNGP) kernel.

Recursive expression for NNGP and NTK. Several previous works [2–5] have shown that the NNGP and NTK can be expressed using the following recursive procedure:

1. For every $x, y \in \mathbb{R}^d$, let $K_\sigma^{(0)}(x,y) := \langle x, y\rangle$ and for every layer $h = 1, \dots, L$, recursively define kernel functions $K_\sigma^{(h)}, \dot{K}_\sigma^{(h)} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ as:

$$K_\sigma^{(h)}(x,y) := \mathop{\mathbb{E}}_{(u,v)\sim\mathcal{N}(0,\boldsymbol{\Lambda}_\sigma^{(h)})}[\sigma(u)\sigma(v)], \qquad \dot{K}_\sigma^{(h)}(x,y) := \mathop{\mathbb{E}}_{(u,v)\sim\mathcal{N}(0,\boldsymbol{\Lambda}_\sigma^{(h)})}[\sigma'(u)\sigma'(v)], \tag{3}$$

where the covariance matrix is $\boldsymbol{\Lambda}_\sigma^{(h)} := \begin{bmatrix} K_\sigma^{(h-1)}(x,x) & K_\sigma^{(h-1)}(x,y) \\ K_\sigma^{(h-1)}(y,x) & K_\sigma^{(h-1)}(y,y) \end{bmatrix} \in \mathbb{R}^{2\times 2}$.

2. The depth-L NNGP kernel is $K_\sigma^{(L)}(x,y)$ and the depth-L NTK $\Theta_\sigma^{(L)}$ can be recursively computed as $\Theta_\sigma^{(0)}(x,y) := \langle x, y\rangle$ and

$$\Theta_\sigma^{(h)}(x,y) := \Theta_\sigma^{(h-1)}(x,y) \cdot \dot{K}_\sigma^{(h)}(x,y) + K_\sigma^{(h)}(x,y). \tag{4}$$

At the core of the expression for $\Theta_\sigma^{(L)}$, there is the expectation term over 2-dimensional Gaussian distribution in Equation (3). This expectation term for the case where both diagonal entries of the covariance matrix $\boldsymbol{\Lambda}_\sigma^{(\ell)}$ are equal to one, was previously studied in [46]. We extend this to encompass general symmetric covariance matrices in the following definition.

[1] Throughout the paper, we consider scalar-valued networks without biases for simplicity, but this can be extended to vector-valued networks with biases . We also assume $\|\sigma\|_{\mathcal{N}} = 1$ which does not change our results.

Definition 1 (Dual Activation and Dual Kernel)**.** For a smooth $\sigma : \mathbb{R} \to \mathbb{R}$, we define the *Dual Kernel* of σ as $K_\sigma : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ defined as

$$K_\sigma(x, y) := \underset{w \sim \mathcal{N}(0, \boldsymbol{I}_d)}{\mathbb{E}} [\sigma(\langle w, x \rangle)\sigma(\langle w, y \rangle)] \quad \text{for every } x, y \in \mathbb{R}^d. \tag{5}$$

Equation (5) only depends on bivariate Gaussian random variables $\langle w, x \rangle, \langle w, y \rangle$ where $\mathbb{E}[\langle w, x \rangle^2] = \|x\|_2^2, \mathbb{E}[\langle w, y \rangle^2] = \|y\|_2^2$ and $\mathbb{E}[\langle w, x \rangle \cdot \langle w, y \rangle] = \langle x, y \rangle$. Hence one can look at the dual kernel from a different perspective by choosing a proper covariance matrix. To this end, let $\boldsymbol{\Lambda}_{a,b,c} := \begin{bmatrix} a^2 & abc \\ abc & b^2 \end{bmatrix}$ for every $a, b \in \mathbb{R}_+$ and $c \in [-1, 1]$ and the *Dual Activation* of σ with respect to $\boldsymbol{\Lambda}_{a,b,c}$ is the function $k_\sigma : \mathbb{R}_+ \times \mathbb{R}_+ \times [-1, 1] \to \mathbb{R}$ defined as $k_\sigma(a, b, c) := \mathbb{E}_{(u,v) \sim \mathcal{N}(0, \boldsymbol{\Lambda}_{a,b,c})} [\sigma(u)\sigma(v)]$.

With these definitions in place, the following relationship between dual kernel and activation holds

$$K_\sigma(x, y) = k_\sigma \left(\|x\|_2, \|y\|_2, \frac{\langle x, y \rangle}{\|x\|_2 \|y\|_2} \right). \tag{6}$$

Observe that $K_\sigma(x, y)$ corresponds to the NNGP kernel of a 1-hidden layer neural network with activation σ. For some specific activations, e.g., ReLU, Error function, closed form expressions for their dual activations are known (see Table 1). Hence, one can compute the NTK analytically when dual kernels of the activation and its derivative have a closed form expression. The above also holds for kernels corresponding to convolutional neural networks called CNN-GP [7, 8] and CNTK [9].

3 NNGP and NTK for Smooth Activations

In this section, we focus on the NNGP and NTK for a wide range of smooth activation functions. We first show that a series expansion for the dual kernel can be obtained from that of the activation function, which is a key to NNGP kernel computation. By applying this result to the derivative of the activation function, we can also compute the NTK for the same activation.

3.1 Dual Kernel Computation

Daniely et al. [46] proved that for absolutely continuous $\sigma : \mathbb{R} \to \mathbb{R}$ and any $x, y \in \mathbb{S}^{d-1}$, the dual kernel is equal to $K_\sigma(x, y) = \sum_{j=0}^{\infty} c_j^2 \, j! \cdot \langle x, y \rangle^j$. where $\{c_j\}_{j \geq 0}$ are coefficients of Hermite expansion of σ. We now proceed to generalize this result from $\mathbb{S}^{d-1}$ to entire $\mathbb{R}^d \setminus \{0\}$. First we remark that it can be naturally extended to the dual kernel of *q-homogeneous* activation functions, i.e., $\sigma(at) = |a|^q \, \sigma(t)$ for every $a, t \in \mathbb{R}$, on the entire $\mathbb{R}^d \setminus \{0\}$. For every $x, y \in \mathbb{R}^d \setminus \{0\}$, the corresponding dual kernel is

$$K_\sigma(x, y) = \|x\|_2^q \, \|y\|_2^q \cdot \sum_{j=0}^{\infty} c_j^2 \, j! \cdot \left(\frac{\langle x, y \rangle}{\|x\|_2 \, \|y\|_2} \right)^j. \tag{7}$$

As examples, (leaky) ReLU and rectified polynomials fall into this activation class.

Now suppose that σ is not homogeneous. In particular, we first consider a polynomial activation $\sigma(t) = \sum_{j=0}^{q} a_j t^j$ with coefficients $\{a_j\}_{j=0}^q$. Recall that $K_\sigma(x, y)$ can be obtained by taking the expectation of $\sigma(\langle w, x \rangle)\sigma(\langle w, y \rangle)$ over $w \sim \mathcal{N}(0, \boldsymbol{I}_d)$ for every $x, y \in \mathbb{R}^d \setminus \{0\}$. To make use of Daniely et al. [46]'s result, we factorize the input into its radial and angular part and rewrite the activation by expressing monomials in the Hermite polynomial basis. Formally, let us write monomials in the Hermite basis as $t^i = \sum_{\ell=0}^{i} \mu_{i,\ell} h_\ell(t)$ for some coefficients $\{\mu_{j,i}\}_{i=0}^j$. Then

$$\sigma(\langle w, x \rangle) = \sum_{j=0}^{q} a_j \, \|x\|_2^j \left\langle w, \frac{x}{\|x\|_2} \right\rangle^j = \sum_{i=0}^{q} \left(\sum_{j=i}^{q} \mu_{j,i} \, \|x\|_2^j \, a_j \right) h_i \left(\left\langle w, \frac{x}{\|x\|_2} \right\rangle \right). \tag{8}$$

Then, we can derive the dual kernel of polynomial activation. We further relax a condition on the activation and propose the result below.

Theorem 1. *For a polynomial $\widetilde{\sigma}(t) = \sum_{j=0}^{q} a_j t^j$, the dual kernel of $\widetilde{\sigma}(\cdot)$, as per Definition 1, is*

$$K_{\widetilde{\sigma}}(x,y) := \sum_{\ell=0}^{q} r_{\widetilde{\sigma},\ell}(\|x\|_2)\, r_{\widetilde{\sigma},\ell}(\|y\|_2) \left(\frac{\langle x,y\rangle}{\|x\|_2 \|y\|_2}\right)^{\ell} \tag{9}$$

where $r_{\widetilde{\sigma},\ell}(t) := \sum_{i=0}^{\lfloor \frac{q-\ell}{2} \rfloor} \frac{a_{\ell+2i}(\ell+2i)!}{2^i \cdot i! \cdot \sqrt{\ell!}} t^{2i+\ell}$. Moreover, if an activation function $\sigma : \mathbb{R} \to \mathbb{R}$ satisfies $\|\sigma\|^2_{\mathcal{N}(0,\nu^2)} < \infty$ and $\|\sigma - \widetilde{\sigma}\|^2_{\mathcal{N}(0,\nu^2)} \leq \varepsilon$ for some $\varepsilon > 0$ and $\nu \geq 1$, then for every $x, y \in \mathbb{R}^d$ such that $\|x\|_2, \|y\|_2 \in (0,\nu]$ the following holds

$$|K_\sigma(x,y) - K_{\widetilde{\sigma}}(x,y)| \leq \sqrt{\frac{\nu^2 \cdot \varepsilon \left(6\|\sigma\|^2_{\mathcal{N}(0,\nu^2)} + 4\varepsilon\right)}{\|x\|_2 \|y\|_2}}. \tag{10}$$

The proof of Theorem 1 is provided in Appendix B.2. For non-polynomial activations, one can consider approximating σ with its Hermite or Taylor expansion and then apply Theorem 1. Examples can be found in Appendix B.2. For activation functions that do not have a Taylor expansion but are k-th order differentiable, we show that, using their Hermite expansion, one can obtain a good approximation to the corresponding dual kernel.

Theorem 2. *Given $\sigma : \mathbb{R} \to \mathbb{R}$, suppose that there exists an integer $k \geq 2$ and some $\nu \geq 1$ such that for every $i = 0, \ldots, k$, $\sigma^{(i)}$ is absolutely continuous and $\lim_{t\to\pm\infty} e^{-\frac{t^2}{4}} \sigma^{(i)}(\nu t) = 0$ and moreover $\|\sigma\|^2_{\mathcal{N}(0,\nu^2)} < \infty$ and $\left\|\sigma^{(k)}\right\|^2_{\mathcal{N}(0,\nu^2)} < \infty$. Consider the Hermite expansion coefficients $\{c_j\}_{j\geq 0}$ of function $\sigma(\nu t)$ and denote $\widetilde{\sigma}(t) := \sum_{j=0}^{q} c_j h_j(t/\nu)$. Given $x, y \in \mathbb{R}^d$ with $\|x\|_2, \|y\|_2 \in (0,\nu]$,*

$$|K_\sigma(x,y) - K_{\widetilde{\sigma}}(x,y)| \leq \frac{5\nu^{k+1} \left\|\sigma^{(k)}\right\|_{\mathcal{N}(0,\nu^2)} \max\left(\|\sigma\|_{\mathcal{N}(0,\nu^2)}, \nu^k \left\|\sigma^{(k)}\right\|_{\mathcal{N}(0,\nu^2)}\right)}{\sqrt{\|x\|_2 \|y\|_2 \cdot k \cdot q^{k-1}}}. \tag{11}$$

where $K_\sigma(\cdot,\cdot)$ and $K_{\widetilde{\sigma}}(\cdot,\cdot)$ are dual kernels corresponding to $\sigma(\cdot)$ and $\widetilde{\sigma}(\cdot)$ in Definition 1, respectively. Moreover, for the ReLU activation $\sigma(t) = \max(t,0)$, it holds that

$$|K_\sigma(x,y) - K_{\widetilde{\sigma}}(x,y)| \leq \sqrt{\frac{2\nu^6}{q\|x\|_2 \|y\|_2}}. \tag{12}$$

The proof of Theorem 2 is provided in Appendix B.3. Observe that when the activation is k-th order differentiable and the norms of its derivative and inputs are bounded then the approximation error decreases with $\mathcal{O}(\frac{1}{\sqrt{kq^{k-1}}})$ rate. In Section 5, we empirically evaluate the dual kernel of various activations using Hermite expansion and verify that smooth activations (e.g., Gaussian or sinusoidal) provides much lower approximation errors than non-smooth ones (e.g., ReLU).

3.2 NNGP and NTK Computations

Once dual kernels of σ and σ' or their polynomial approximations are calculated, one can compute (approximate) NNGP and NTK using Theorem 1 or Theorem 2 and the recursion in Equation (4). However, there are scenarios where we are only given the dual kernel and the corresponding activation or derivative of the activation is unknown to us. For example, Shankar et al. [54] devised a normalized Gaussian kernel defined as

$$K_G(x,y) = \|x\|_2 \|y\|_2 \exp\left(\frac{\langle x,y\rangle}{\|x\|_2 \|y\|_2} - 1\right), \tag{13}$$

and reported that NNGP with this dual kernel performs better than the ReLU NTK by showing promising results on various tasks. Note that, recovering the activation from K_G is non-trivial. From the dual kernel perspective, the activation should be 1-homogeneous and its Hermite series expansion is of form $\sum_{j=0}^{\infty} \frac{\pm 1}{j!} h_j(t)$ and it is generally unknown how to choose the sign pattern on coefficients of this series that would satisfy homogeneity constraint. Instead of trying to recover the activation from dual kernel, we show how to directly derive the dual kernel of derivative of activation without knowing the activation.

Theorem 3. *Given a differentiable activation function $\sigma : \mathbb{R} \to \mathbb{R}$ which satisfies $|\sigma(t)| \leq C_1 \exp\left(\frac{t^2}{4.1\nu^2}\right)$, $|\sigma'(t)| \leq C_2 \exp\left(\frac{t^2}{4.1\nu^2}\right)$, $\|\sigma\|^2_{\mathcal{N}(0,\nu^2)} < \infty$ and $\|\sigma''\|^2_{\mathcal{N}(0,\nu^2)} < \infty$ for some $\nu \geq 1$ and constants C_1, C_2, the following holds for any $x, y \in \mathbb{R}^d$ with $\|x\|_2, \|y\|_2 \in (0, \nu]$ and $|\langle x, y\rangle| < \|x\|_2 \|y\|_2$:*

$$K_{\sigma'}(x, y) = \frac{1}{\|x\|_2 \|y\|_2} \frac{\partial}{\partial c} k_\sigma \left(\|x\|_2, \|y\|_2, c\right) \Bigg|_{c=\frac{\langle x,y\rangle}{\|x\|_2\|y\|_2}}. \tag{14}$$

Additionally, if $\frac{\partial}{\partial c} k_\sigma(\cdot, \cdot, c)$ is continuous at $c = \pm 1$ then Equation (14) holds for x, y such that $|\langle x, y\rangle| = \|x\|_2 \|y\|_2$.

The proof of Theorem 3 is provided in Appendix B.5. Our result is more general compared to [49] where the previous work assumes that the Hermite expansion of given activation should converge and $\|x\|_2 = \|y\|_2$. Applying Theorem 3 to Equation (13) provides that $\dot{K}_G(x, y) = \exp\left(\frac{\langle x,y\rangle}{\|x\|_2\|y\|_2} - 1\right)$ hence one can compute the NTK function even if the corresponding activation is unknown. In the previous work [54], only "NNGP" performances of the normalized Gaussian kernel were reported.

Moreover, with Theorem 3, only the knowledge of dual activation suffices to compute both NNGP and NTK. For example, while dual activation (thus NNGP) of GeLU was known in Tsuchida et al. [48], $k_{\sigma'}$ was not derived explicitly. Theorem 3 provides a simple way to compute $k_{\sigma'}$ (given in Equation (126)) via automatic differentiation, without requiring to take the expectation under multivariate Gaussian distribution or computing derivatives by hand. This is implemented in `stax_extensions.Elementwise` in our code supplement. Our method allows to omit the enitre effort, lines of code, and potential mistakes in deriving and implementing the NTK.

3.3 Gauss-Hermite Quadrature

One simple approach to obtain dual activation function for general activation functions without closed form expressions is to evaluate the expectation of under the $2d$ Gaussian distribution as numerical integration. This can be efficiently done by Gauss-Hermite quadrature

$$k_\sigma(a, b, c) \approx \frac{1}{\pi} \sum_{i=1}^{q} \sum_{j=1}^{q} w_i w_j \left[\sigma(\sqrt{2}a x_i) \cdot \sigma(\sqrt{2}bc x_i + \sqrt{2}b\sqrt{1-c^2} x_j)\right] \tag{15}$$

where (x_i, w_i), correspond to i-th root of degree q Hermite polynomial $h_i(x)$ and associated weights [55] $w_i = \frac{q!\sqrt{\pi}}{q^2(h_{q-1}(\sqrt{2}x_i))^2}$. See Appendix E for the derivation of the quadrature formula.

For smooth activation functions errors will quickly go down as q increases by Theorem 2. We use this method to compute approximate (non-sketched) kernels for general activation functions in Figure 3 and Figure 4. It is implemented as `stax_extensions.ElementwiseNumerical` in our code.

4 Approximating Neural Kernels via Sketching

Although using our Theorem 1, Theorem 2, and Theorem 3, one can analytically compute NTK for general activation functions, computing all entries in the NTK kernel matrix requires massive amount of resources, i.e., $\Omega(n^2(d + Lq^2))$ runtime and $\Omega(n^2)$ memory for datasets with n points in $\mathbb{R}^d$. This becomes even more expensive for CNTK, where its runtime can be $\Omega((nd_1d_2)^2(c + Lq^2))$[2] for n of images with size $d_1 \times d_2 \times c$. To avoid quadratic complexities, we adopt a fast and efficient feature map construction via randomized sketching [41] for both NTK and NNGP, i.e.,

$$\Theta_\sigma^{(L)}(x, y) \approx \left\langle \psi^{(L)}(x), \psi^{(L)}(y) \right\rangle, \quad K_\sigma^{(L)}(x, y) \approx \left\langle \phi^{(L)}(x), \phi^{(L)}(y) \right\rangle. \tag{16}$$

The previous approach was only applicable for the ReLU activation but we establish more general scheme based on our new results for dual kernel approximation.

[2] This is assuming Hermite expansion degree q, when exact expression is known q^2 is constant.

Algorithm 1 Subspace Embedding of Homogeneous NNGP and NTK

1: **input**: $x \in \mathbb{R}^d$, depth L, sketching dimension m, polynomial $\widetilde{\kappa}(t) = \sum_{j=0}^{q} a_j t^j$ with $a_j \in \mathbb{R}_+$
2: calculate the polynomial $P^{(L)}(t) = \widetilde{\kappa}^{\circ L}(t) = \sum_{j=0}^{q^L} b_j t^j$ with coefficients $b_j \in \mathbb{R}_+$
3: calculate the polynomial $R^{(L)}(t) = \sum_{h=0}^{L} \widetilde{\kappa}^{\circ h}(t) \cdot \prod_{i=h}^{L-1} \widetilde{\kappa}' \circ \widetilde{\kappa}^{\circ i}(t) = \sum_{j=0}^{p} c_j t^j$ with coefficients $c_j \in \mathbb{R}_+$ and degree $p = q^{\mathcal{O}(L)}$
4: for $\ell = 0, \ldots, p$, let $Q^\ell \in \mathbb{R}^{m \times d^\ell}$ be a degree-ℓ PolySketch (See Appendix A)
5: for every $\ell = 0, \ldots, p$, $u^\ell \leftarrow Q^\ell \left(\frac{x}{\|x\|_2}\right)^{\otimes \ell}$
6: construct $\phi^{(L)}(x) \leftarrow \|x\|_2 \cdot \bigoplus_{j=0}^{q^L} \sqrt{b_j} u^j$ and $\psi^{(L)}(x) \leftarrow \|x\|_2 \cdot \bigoplus_{j=0}^{p} \sqrt{c_j} u^j$
7: **return** $\phi^{(L)}(x)$ (NNGP embedding), $\psi^{(L)}(x)$ (NTK embedding)

Subspace embedding for homogeneous dual kernels. We provide a subspace embedding for NNGP and NTK matrices with near input-sparsity runtime and near-optimal target dimension which applies to any *homogeneous* dual activation functions with rapidly convergent Taylor expansion. More specifically, we call a dual kernel K_σ homogeneous if there exists a positive definite dot-product kernel function $\kappa : [-1, 1] \to [-1, 1]$ such that,

$$K_\sigma(x, y) = \|x\|_2 \|y\|_2 \cdot \kappa\left(\frac{\langle x, y\rangle}{\|x\|_2 \|y\|_2}\right). \tag{17}$$

For such homogeneous dual kernels, the NTK and NNGP take a similar homogeneous form. In fact, one can show by induction that when the dual kernel is in form of Equation (17), the depth-L NNGP function defined in Equation (3) is equal to the following for any positive integer L,

$$K_\sigma^{(L)}(x, y) = \|x\|_2 \|y\|_2 \cdot \kappa^{\circ L}\left(\frac{\langle x, y\rangle}{\|x\|_2 \|y\|_2}\right), \tag{18}$$

where $\kappa^{\circ L}$ denoted the L-fold composition of function κ. Furthermore, if κ has a derivative $\kappa' : [-1, 1] \to [-1, 1]$, using Theorem 3, there exists a depth-L NTK for this dual kernel, equal to

$$\Theta_\sigma^{(L)}(x, y) = \|x\|_2 \|y\|_2 \cdot \sum_{h=0}^{L} \kappa^{\circ h}(t) \cdot \prod_{i=h}^{L-1} \kappa' \circ \kappa^{\circ i}(t) \Bigg|_{t = \frac{\langle x, y\rangle}{\|x\|_2 \|y\|_2}}, \tag{19}$$

where we use the convention that $\kappa^{\circ 0}(t) = t$. Therefore, if $\kappa(\cdot)$ can be tightly approximated by a low-degree polynomial, then the NNGP and NTK functions can also be tightly approximated by low-degree polynomials. Thus, by applying PolySketch, which is a norm-preserving dimensionality reduction that can be applied to the tensor product of multiple vectors very quickly [56], to the polynomial approximations to these kernels, we can spectrally approximate the NNGP and NTK kernel matrices. For details on PolySketch see Appendix A. We provide the details of this procedure in Algorithm 1 and prove the correctness and runtime of our procedure in Theorem 4.

Theorem 4 (Homogeneous NTK Embedding)**.** *Suppose that the dual kernel K_σ is homogeneous as per Equation (17). Also suppose $\widetilde{\kappa}(t)$ is a degree-q polynomial with non-negative coefficients that satisfies* (**1**) $\max_{t \in [-1,1]} |\widetilde{\kappa}(t) - \kappa(t)| \leq \frac{1}{\mathrm{poly}(n)}$ *and* $\max_{t \in [-1,1]} |\widetilde{\kappa}'(t) - \kappa'(t)| \leq \frac{1}{\mathrm{poly}(n)}$, (**2**) $\max_{|t| \leq 1 + \frac{1}{\mathrm{poly}(n)}} |\widetilde{\kappa}(t + \gamma) - \widetilde{\kappa}(t)| \leq \frac{1}{\mathrm{poly}(n)}$ *and* $\max_{|t| \leq 1 + \frac{1}{\mathrm{poly}(n)}} |\widetilde{\kappa}'(t + \gamma) - \widetilde{\kappa}'(t)| \leq \frac{1}{\mathrm{poly}(n)}$ *for any* $|\gamma| \leq \frac{1}{\mathrm{poly}(n)}$. *Then for any integer $L \geq 1$, any $\varepsilon, \lambda \geq \frac{1}{\mathrm{poly}(n)}$, and any dataset $\boldsymbol{X} \in \mathbb{R}^{d \times n}$ with $\|\boldsymbol{X}\|_F \leq \mathrm{poly}(n)$, if $\boldsymbol{K}_{\mathrm{ntk}} \in \mathbb{R}^{n \times n}$ is the depth-L NTK kernel matrix on this dataset, there exists $m = \mathcal{O}\left(\frac{s_\lambda(\boldsymbol{K}_{\mathrm{ntk}})}{\varepsilon^2} \cdot \mathrm{poly}\left(q^L, \log n\right)\right)$ such that the output $\psi^{(L)}(\boldsymbol{X}) \in \mathbb{R}^{m \times n}$ of Algorithm 1 satisfies with probability at least $1 - \frac{1}{\mathrm{poly}(n)}$*

$$(1 - \varepsilon)(\boldsymbol{K}_{\mathrm{ntk}} + \lambda \boldsymbol{I}_n) \preceq \psi^{(L)}(\boldsymbol{X})^\top \psi^{(L)}(\boldsymbol{X}) + \lambda \boldsymbol{I}_n \preceq (1 + \varepsilon)(\boldsymbol{K}_{\mathrm{ntk}} + \lambda \boldsymbol{I}_n). \tag{20}$$

Moreover, the runtime of Algorithm 1 is $\mathcal{O}\left(\mathrm{poly}\left(q^L, \log n\right) \cdot \varepsilon^{-2} \cdot (s_\lambda(\boldsymbol{K}_{\mathrm{ntk}}) \cdot n + \mathrm{nnz}(\boldsymbol{X}))\right)$.

We prove this theorem in Appendix C. As an example, let us apply Theorem 4 on the normalized Gaussian kernel K_G defined in Equation (13), which is homogeneous. The dot-product factor corresponding to this dual kernel is $\kappa(t) = \exp(t - 1)$. The truncated Taylor series of this

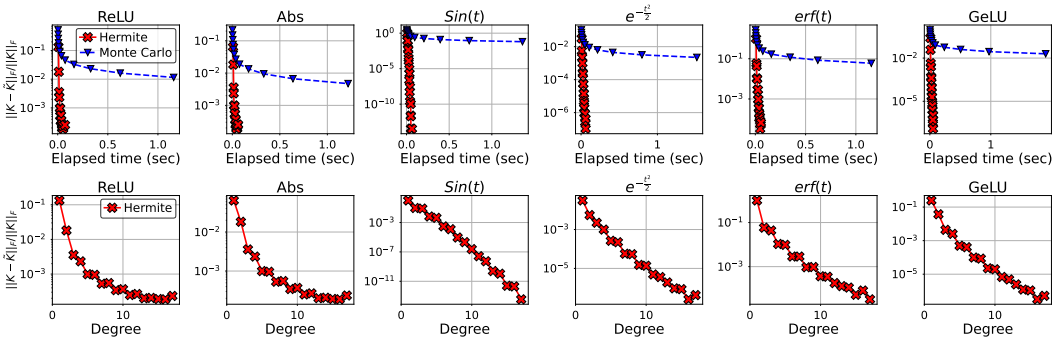

Figure 1: Relative errors of dual kernel approximations via the truncated Hermite expansion and Monte Carlo estimation under synthetic dataset with $n = 1{,}000$, $d = 256$.

function is $\widetilde{\kappa}(t) = \sum_{j=0}^{q} \frac{t^j}{e \cdot j!}$. If $q = \Omega(\log n)$ then it can be verified that the polynomial $\widetilde{\kappa}(t)$ satisfies the preconditions of Theorem 4. Therefore, one can invoke Algorithm 1 to get a subspace embedding for the NTK kernel matrix corresponding to the normalized Gaussian dual kernel K_G in $\mathcal{O}\left(\varepsilon^{-2} \cdot (s_\lambda(\boldsymbol{K}_{\mathrm{ntk}}) \cdot n + \mathrm{nnz}(\boldsymbol{X})) \cdot \mathrm{poly}\left(\log^L n\right)\right)$ time and with a target dimension of $m = \mathcal{O}\left(\varepsilon^{-2} \cdot s_\lambda(\boldsymbol{K}_{\mathrm{ntk}}) \cdot \mathrm{poly}\left(\log^L n\right)\right)$. For any constant number of layers, L, this runtime and target dimension is is optimal up to $\mathrm{poly}\left(\log n\right)$ factors. The implementation of our sketching algorithm is available at `https://github.com/insuhan/ntk_activations`.

5 Experiments

In this section, we perform experiments with the proposed neural kernels based on our dual kernel approximation. All experiments run using a single A100 GPU machine.

Kernel approximation. We first benchmark our algorithm to approximate the dual kernel matrix. We use ReLU, Abs (i.e., $\sigma(t) = |t|$), sin, Gaussian, erf and GeLU activations and approximate them by their Hermite expansion where degree changes from $q = 1$ to 20. We randomly generate $n = 1{,}000$ of 256-dimensional inputs where each entry is i.i.d. drawn from $\mathcal{N}(0, 1/\sqrt{256})$. We also compare our approach to the Monte Carlo estimation of dual kernel, i.e., $K_\sigma(x, y) \approx \frac{1}{m} \sum_{i=1}^{m} \sigma(\langle w_i, x\rangle)\sigma(\langle w_i, y\rangle)$ where $\{w_i\}_{i=1}^m$ are i.i.d. standard Gaussian vectors. In Figure 1, we plot relative errors of the Frobenius norm of kernel approximations in terms of wall-clock times (**top**) and polynomial degree (**bottom**). We run 10 independent trials and evaluate the average approxmation errors. We observe that our approximation with Hermite expansion outperforms the Monte Carlo method for all activations we used. In particular, sin and Gaussian are well approximated because they are smooth and norms of their derivatives are bounded with respect to the normal measure.

Performance on CIFAR-10 classification. We also benchmark the proposed CNTK approximating via sketching algorithm. We perform CIFAR-10 classification [57] by solving the ridge regression problem. The image classes are converted into 10-dimensional one-hot vectors and inputs are pre-processed with regularized ZCA [54, 58]. We report the best test accuracy among 20 choices of ridge parameters in $\{10^{-10+\frac{12}{19}i} \mid i = 0, 1, \ldots, 19\}$. We extract CNTK features of a 5-layer convolutional neural network (known as Myrtle5 [54]) without pooling by setting degree $q = 8$ and explore feature dimension $m = \{2^9, \ldots, 2^{14}\}$ and homogeneous dual kernels including ReLU, ABReLU activations as well as deep normalized Gaussian kernels with 2 scaling factors. See Appendix G for more details. In Figure 2, the test accuracy of neural kernels (**left**) and the corresponding their dual activations (**right**) are plotted. The dual activation of

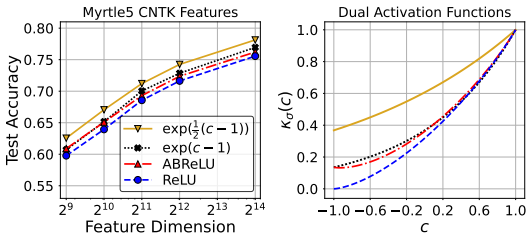

Figure 2: Test accuracy of CIFAR-10

ABReLU is very similar to the normalized Gaussian without scaling and their test performances are also comparable. We observe that the scaled normalized Gaussian shows the best performance which achieves 78.13% while the ReLU CNTK features [41] shows 75.56% with the same runtime. This is because the coefficients decay of the normalized Gaussian is faster than that of the ReLU, which leads to a lower approximation error of sketching algorithm. We also perform comparison among different activation functions in neural kernels in Appendix E.

Speedup. We observe that the exact CNTK of Myrtle-5 constructs a kernel matrix of size $60{,}000 \times 60{,}000$ and achieves 86-87% test accuracy. However, this requires approximately 151 GPU hours. Under the same setting, our CNTK features for the normalized Gaussian kernel take about 1.4 GPU hours, i.e. a $106\times$ speedup. If we use less training data to construct $20{,}000 \times 20{,}000$ kernel matrix, the accuracy is about 77% accuracy and the runtime is 16.8 GPU hours in which our approximation is still $12\times$ faster without loss of accuracy. We believe such acceleration through our methods open the door to using neural kernels in a wide range of research domains.

6 Discussion

In this work, we introduced methods to efficiently compute neural kernels for general activations. As activation functions play an important role in modern neural network architectures, we hope our work could empower researchers to explore properties of activations in a more principled way. We are excited with sketching method's compute efficiency by orders of magnitude on highly performant neural kernels to open up applications in dataset distillation [29] or uncertainty critical problems [59] such as autonomous driving, healthcare and science.

Acknowledgements

Amir Zandieh was supported by the Swiss NSF grant No. P2ELP2_195140. Insu Han and Amin Karbasi acknowledge funding in direct support of this work from NSF (IIS-1845032), ONR (N00014-19-1-2406), and the AI Institute for Learning-Enabled Optimization at Scale (TILOS). We thank Timothy Nguyen and Jeffrey Pennington for discussions and feedback on the project.

References

[1] Radford M. Neal. Priors for infinite networks. *University of Toronto*, 1994.

[2] Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep Neural Networks as Gaussian Processes. In *International Conference on Learning Representations (ICLR)*, 2018.

[3] Alexander G. de G. Matthews, Jiri Hron, Mark Rowland, Richard E. Turner, and Zoubin Ghahramani. Gaussian Process Behaviour in Wide Deep Neural Networks. In *International Conference on Learning Representations (ICLR)*, 2018.

[4] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Neural Information Processing Systems (NeurIPS)*, 2018.

[5] Jaehoon Lee, Lechao Xiao, Samuel Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. In *Neural Information Processing Systems (NeurIPS)*, 2019.

[6] Lechao Xiao. Eigenspace restructuring: a principle of space and frequency in neural networks. In *Conference on Learning Theory (COLT)*, 2022.

[7] Roman Novak, Lechao Xiao, Jaehoon Lee, Yasaman Bahri, Greg Yang, Jiri Hron, Daniel A. Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. Bayesian Deep Convolutional Networks with Many Channels are Gaussian Processes. In *International Conference on Learning Representations (ICLR)*, 2019.

[8] Adrià Garriga-Alonso, Laurence Aitchison, and Carl Edward Rasmussen. Deep convolutional networks as shallow Gaussian processes. In *International Conference on Learning Representations (ICLR)*, 2019.

[9] Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Russ R Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. In *Neural Information Processing Systems (NeurIPS)*, 2019.

[10] Jiri Hron, Yasaman Bahri, Jascha Sohl-Dickstein, and Roman Novak. Infinite attention: NNGP and NTK for deep attention networks. In *International Conference on Machine Learning (ICML)*, 2020.

[11] Simon S Du, Kangcheng Hou, Russ R Salakhutdinov, Barnabas Poczos, Ruosong Wang, and Keyulu Xu. Graph Neural Tangent Kernel: Fusing Graph Neural Networks with Graph Kernels. In *Neural Information Processing Systems (NeurIPS)*. 2019.

[12] Greg Yang. Scaling Limits of Wide Neural Networks with Weight Sharing: Gaussian Process Behavior, Gradient Independence, and Neural Tangent Kernel Derivation. *arXiv preprint arXiv:1902.04760*, 2019.

[13] Sanjeev Arora, Nadav Cohen, and Elad Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization. In *International Conference on Machine Learning (ICML)*, 2018.

[14] Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning (ICML)*, 2019.

[15] Mert Pilanci and Tolga Ergen. Neural networks are convex regularizers: Exact polynomial-time convex optimization formulations for two-layer networks. In *International Conference on Machine Learning (ICML)*, 2020.

[16] Lechao Xiao, Jeffrey Pennington, and Samuel Schoenholz. Disentangling trainability and generalization in deep neural networks. In *International Conference on Machine Learning (ICML)*, 2020.

[17] Behnam Neyshabur, Zhiyuan Li, Srinadh Bhojanapalli, Yann LeCun, and Nathan Srebro. The role of over-parametrization in generalization of neural networks. In *International Conference on Learning Representations (ICLR)*, 2019.

[18] Sanjeev Arora, Simon Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *International Conference on Machine Learning (ICML)*, 2019.

[19] Yuan Cao and Quanquan Gu. Generalization bounds of stochastic gradient descent for wide and deep neural networks. In *Neural Information Processing Systems (NeurIPS)*, 2019.

[20] Colin Wei, Jason D Lee, Qiang Liu, and Tengyu Ma. Regularization matters: Generalization and optimization of neural nets vs their induced kernel. *Neural Information Processing Systems (NeurIPS)*, 2019.

[21] Wei Hu, Zhiyuan Li, and Dingli Yu. Simple and effective regularization methods for training on noisily labeled data with generalization guarantee. In *International Conference on Learning Representations (ICLR)*, 2020.

[22] Arthur Jacot, Berfin Simsek, Francesco Spadaro, Clément Hongler, and Franck Gabriel. Implicit regularization of random feature models. In *International Conference on Machine Learning (ICML)*, 2020.

[23] Elvis Dohmatob. Fundamental tradeoffs between memorization and robustness in random features and neural tangent regimes. *arXiv preprint arXiv:2106.02630*, 2021.

[24] Hamed Hassani and Adel Javanmard. The curse of overparametrization in adversarial training: Precise analysis of robust generalization for random features regression. *arXiv preprint arXiv:2201.05149*, 2022.

[25] Sanjeev Arora, Simon S Du, Zhiyuan Li, Ruslan Salakhutdinov, Ruosong Wang, and Dingli Yu. Harnessing the Power of Infinitely Wide Deep Nets on Small-data Tasks. In *International Conference on Learning Representations (ICLR)*, 2019.

[26] Daniel S Park, Jaehoon Lee, Daiyi Peng, Yuan Cao, and Jascha Sohl-Dickstein. Towards nngp-guided neural architecture search. *arXiv preprint arXiv:2011.06006*, 2020.

[27] Wuyang Chen, Xinyu Gong, and Zhangyang Wang. Neural Architecture Search on ImageNet in Four GPU Hours: A Theoretically Inspired Perspective. In *International Conference on Learning Representations (ICLR)*, 2021.

[28] Timothy Nguyen, Zhourong Chen, and Jaehoon Lee. Dataset Meta-Learning from Kernel Ridge-Regression. In *International Conference on Learning Representations (ICLR)*, 2021.

[29] Timothy Nguyen, Roman Novak, Lechao Xiao, and Jaehoon Lee. Dataset distillation with infinitely wide convolutional networks. In *Neural Information Processing Systems (NeurIPS)*, 2021.

[30] Baihe Huang, Xiaoxiao Li, Zhao Song, and Xin Yang. Fl-ntk: A neural tangent kernel-based framework for federated learning analysis. In *International Conference on Machine Learning (ICML)*, 2021.

[31] Yufan Zhou, Zhenyi Wang, Jiayi Xian, Changyou Chen, and Jinhui Xu. Meta-learning with neural tangent kernels. *International Conference on Learning Representations (ICLR)*, 2021.

[32] Chia-Hung Yuan and Shan-Hung Wu. Neural tangent generalization attacks. In *International Conference on Machine Learning (ICML)*, 2021.

[33] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[34] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Association for Computational Linguistics (NAACL)*, 2018.

[35] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.

[36] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Neural Information Processing Systems (NeurIPS)*, 2020.

[37] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Neural Information Processing Systems (NeurIPS)*, 2020.

[38] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Neural Information Processing Systems (NeurIPS)*, 2020.

[39] Cihang Xie, Mingxing Tan, Boqing Gong, Alan Yuille, and Quoc V Le. Smooth adversarial training. *arXiv preprint arXiv:2006.14536*, 2020.

[40] Roman Novak, Jascha Sohl-Dickstein, and Samuel S. Schoenholz. Fast Finite Width Neural Tangent Kernel. In *International Conference on Machine Learning*, 2022. URL `https://github.com/google/neural-tangents`.

[41] Amir Zandieh, Insu Han, Haim Avron, Neta Shoham, Chaewon Kim, and Jinwoo Shin. Scaling Neural Tangent Kernels via Sketching and Random Features. In *Neural Information Processing Systems (NeurIPS)*, 2021.

[42] Roman Novak, Lechao Xiao, Jiri Hron, Jaehoon Lee, Alexander A. Alemi, Jascha Sohl-Dickstein, and Samuel S. Schoenholz. Neural Tangents: Fast and Easy Infinite Neural Networks in Python. In *International Conference on Learning Representations (ICLR)*, 2020. URL `https://github.com/google/neural-tangents`.

[43] Christopher Williams. Computing with infinite networks. In *Neural Information Processing Systems (NeurIPS)*, 1996.

[44] Youngmin Cho and Lawrence Saul. Kernel methods for deep learning. In *Neural Information Processing Systems (NeurIPS)*, 2009.

[45] Ali Rahimi and Benjamin Recht. Random Features for Large-Scale Kernel Machines. In *Neural Information Processing Systems (NeurIPS)*, 2009.

[46] Amit Daniely, Roy Frostig, and Yoram Singer. Toward deeper understanding of neural networks: The power of initialization and a dual view on expressivity. In *Neural Information Processing Systems (NeurIPS)*, 2016.

[47] Cosme Louart, Zhenyu Liao, Romain Couillet, et al. A random matrix approach to neural networks. *The Annals of Applied Probability*, 2018.

[48] Russell Tsuchida, Tim Pearce, Chris van der Heide, Fred Roosta, and Marcus Gallagher. Avoiding Kernel Fixed Points: Computing with ELU and GELU Infinite Networks. *Conference on Artificial Intelligence (AAAI)*, 2021.

[49] James B Simon, Sajant Anand, and Michael R DeWeese. Reverse Engineering the Neural Tangent Kernel. *arXiv preprint arXiv:2106.03186*, 2021.

[50] Russell Tsuchida, Fred Roosta, and Marcus Gallagher. Invariance of weight distributions in rectified MLPs. In *International Conference on Learning Representations (ICLR)*, 2018.

[51] Russell Tsuchida, Fred Roosta, and Marcus Gallagher. Richer priors for infinitely wide multi-layer perceptrons. *arXiv preprint arXiv:1911.12927*, 2019.

[52] Julien Mairal, Piotr Koniusz, Zaid Harchaoui, and Cordelia Schmid. Convolutional kernel networks. In *Neural Information Processing Systems (NeurIPS)*, 2014.

[53] Tim Pearce, Russell Tsuchida, Mohamed Zaki, Alexandra Brintrup, and Andy Neely. Expressive priors in Bayesian neural networks: Kernel combinations and periodic functions. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2020.

[54] Vaishaal Shankar, Alex Fang, Wenshuo Guo, Sara Fridovich-Keil, Jonathan Ragan-Kelley, Ludwig Schmidt, and Benjamin Recht. Neural kernels without tangents. In *International Conference on Machine Learning (ICML)*, 2020.

[55] Milton Abramowitz, Irene A Stegun, and Robert H Romer. Handbook of mathematical functions with formulas, graphs, and mathematical tables, 1988.

[56] Thomas D Ahle, Michael Kapralov, Jakob BT Knudsen, Rasmus Pagh, Ameya Velingker, David P Woodruff, and Amir Zandieh. Oblivious sketching of high-degree polynomial kernels. In *Symposium on Discrete Algorithms (SODA)*, 2020.

[57] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

[58] Jaehoon Lee, Samuel Schoenholz, Jeffrey Pennington, Ben Adlam, Lechao Xiao, Roman Novak, and Jascha Sohl-Dickstein. Finite versus infinite neural networks: an empirical study. *Neural Information Processing Systems (NeurIPS)*, 2020.

[59] Ben Adlam, Jaehoon Lee, Lechao Xiao, Jeffrey Pennington, and Jasper Snoek. Exploring the Uncertainty Properties of Neural Networks' Implicit Priors in the Infinite-Width Limit. In *International Conference on Learning Representations (ICLR)*, 2021.

[60] Ryan O'Donnell. *Analysis of boolean functions*. Cambridge University Press, 2014.

[61] Shuhuang Xiang. Asymptotics on Laguerre or Hermite polynomial expansions and their applications in Gauss quadrature. *Journal of Mathematical Analysis and Applications*, 2012.

[62] Achim Klenke. *Probability theory: a comprehensive course*. Springer Science & Business Media, 2013.

[63] Gene H Golub and John H Welsch. Calculation of Gauss quadrature rules. *Mathematics of computation*, 1969.

[64] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

[65] Peter Jäckel. A note on multivariate Gauss-Hermite quadrature. *London: ABN-Amro. Re*, 2005.

[66] Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. Rectifier nonlinearities improve neural network acoustic models. In *International Conference on Learning Representations (ICLR)*, 2013.

[67] Ben Adlam, Jake Levinson, and Jeffrey Pennington. A random matrix perspective on mixtures of nonlinearities in high dimensions. In *Conference on Artificial Intelligence and Statistics (AISTATS)*, 2022.

[68] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *International Conference on Learning Representations (ICLR)*, 2020.

[69] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[70] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). In *International Conference on Learning Representations (ICLR)*, 2016.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [No]
   (c) Did you discuss any potential negative societal impacts of your work? [No]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] All detailed proofs in the paper are provided in supplement.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We provide anonymous code of some of our implementation.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Sketching Preliminaries

The POLYSKETCH algorithm is a norm-preserving dimensionality reduction that can be applied to the tensor product of multiple vectors very quickly [56], i.e., for any $v_1, \ldots, v_q \in \mathbb{R}^d$, there exists a randomized mapping $Q^q : \mathbb{R}^{d^q} \to \mathbb{R}^m$ which satisfies that

$$\|Q^q(v_1 \otimes \cdots \otimes v_p)\|_2 \approx \|v_1 \otimes \cdots \otimes v_p\|_2$$

with high probability and $Q^q(v_1 \otimes \cdots \otimes v_p)$ can be computed very fast. Here, sketching dimension m is a trade-off parameter between runtime and accuracy. Algorithm 2 describes the pseudo-code of POLYSKETCH and Theorem 5 summarizes Theorems 1.2 and 1.3 of [56] which guarantees spectral approximation of output of POLYSKETCH.

Algorithm 2 POLYSKETCH [56]

input: $x \in \mathbb{R}^d$, degree q, sketch dimension m, SRHT instances $\{S_j^0 : \mathbb{R}^d \to \mathbb{R}^m\}_{j=1}^q$ and $\{S_j^i : \mathbb{R}^{m^2} \to \mathbb{R}^m \mid j = 1, \ldots, 2^{\lceil \log_2 q \rceil - i}, i = 1, \ldots, \lceil \log_2 q \rceil\}$
let $\overline{q} \leftarrow 2^{\lceil \log_2 q \rceil}$
for every $j = 1, \ldots, q$, let $y_j^0 \leftarrow S_j^0 \cdot x$
for every $j = q+1, \ldots, \overline{q}$, let $y_j^0 \leftarrow S_j^0 \cdot e_1$ where $e_1 \in \mathbb{R}^d$ is the first column vector of $\boldsymbol{I}_d$
for $i = 1, \ldots, \log_2 \overline{q}$
  for $j = 1, \ldots, \overline{q}/2^i$
    compute $y_j^i \leftarrow S_j^i \cdot \left(y_{2j-1}^{i-1} \otimes y_{2j}^{i-1}\right)$
return $z = y_1^{\log_2 \overline{q}}$

Algorithm 3 Subsampled Randomized Hadamard Transform (SRHT)

input: $x \in \mathbb{R}^d$, dimension m, random signs $s \in \{+1, -1\}^d$, random indices $b \in \{1, \ldots, d\}^m$
let $y \leftarrow [x_1 s_1, x_2 s_2, \ldots, x_d s_d]$
compute $z \leftarrow \mathrm{FFT}(y)$
return $\frac{1}{\sqrt{m}} [z_{b_1}, \ldots, z_{b_m}]$

Theorem 5 (POLYSKETCH)**.** *For every integers $p, d \geq 1$ and every $\varepsilon, \delta > 0$, there exists a distribution on random matrices $Q^p \in \mathbb{R}^{m \times d^p}$, called degree p* POLYSKETCH *such that* **(1)** *for some $m = \mathcal{O}\left(\frac{p}{\varepsilon^2} \log^3 \frac{1}{\varepsilon\delta}\right)$ and any $y \in \mathbb{R}^{d^p}$, $\Pr\left[\|Q^p y\|_2^2 \in (1 \pm \varepsilon)\|y\|_2^2\right] \geq 1 - \delta$;* **(2)** *for any $x \in \mathbb{R}^d$, the total time to compute $Q^p x^{\otimes p}$ is $\mathcal{O}\left(pm \log m + \frac{p^{3/2}}{\varepsilon} \log \frac{1}{\delta} \operatorname{nnz}(x)\right)$;* **(3)** *for any collection of vectors $v_1, \ldots, v_p \in \mathbb{R}^d$, the time to compute $Q^p(v_1 \otimes \cdots \otimes v_p)$ is bounded by $\mathcal{O}\left(pm \log m + \frac{p^{3/2}}{\varepsilon} d \log \frac{1}{\delta}\right)$;* **(4)** *for any $\lambda > 0$ and any matrix $\boldsymbol{A} \in \mathbb{R}^{d^p \times n}$, where the statistical dimension of $\boldsymbol{A}^\top \boldsymbol{A}$ is s_λ, there exists some $m = \mathcal{O}\left(\frac{p^4 s_\lambda}{\varepsilon^2} \log^3 \frac{n}{\varepsilon\delta}\right)$ such that,*

$$\Pr\left[(1-\varepsilon)\left(\boldsymbol{A}^\top \boldsymbol{A} + \lambda \boldsymbol{I}_n\right) \preceq (Q^p \boldsymbol{A})^\top (Q^p \boldsymbol{A}) + \lambda \boldsymbol{I}_n \preceq (1+\varepsilon)\left(\boldsymbol{A}^\top \boldsymbol{A} + \lambda \boldsymbol{I}_n\right)\right] \geq 1 - \delta. \quad (21)$$

B Proofs

B.1 Properties of Hermite Polynomials

We first introduce that Hermite polynomials can be used as the random feature of monomial kernels for inputs on the unit sphere, which will be used in our analysis.

Proposition 1. *For $x, y \in \mathbb{S}^{d-1}$, it holds that*

$$\mathop{\mathbb{E}}_{w \sim \mathcal{N}(0, \boldsymbol{I}_d)} \left[h_\ell(\langle w, x \rangle)\, h_m(\langle w, y \rangle)\right] = \ell! \langle x, y \rangle^\ell \cdot \mathbb{1}_{\{\ell = m\}}. \quad (22)$$

Proof of Proposition 1: Let $a := \langle w, x \rangle, b := \langle w, y \rangle$ then $\mathbb{E}_w[a] = \mathbb{E}_w[b] = 0$ and $\mathrm{Cov}(a, b) = \mathbb{E}[ab] = \langle x, y \rangle$. Hence, we have that

$$\mathop{\mathbb{E}}_{w \sim \mathcal{N}(0, \boldsymbol{I}_d)} \left[h_\ell(\langle w, x \rangle)\, h_m(\langle w, y \rangle)\right] = \mathop{\mathbb{E}}_{(a,b) \sim \mathcal{N}(0, \Sigma)} \left[h_\ell(a)\, h_m(b)\right] \quad (23)$$

where

$$\Sigma = \begin{bmatrix} \|x\|_2^2 & \langle x, y\rangle \\ \langle x, y\rangle & \|y\|_2^2 \end{bmatrix} = \begin{bmatrix} 1 & \langle x, y\rangle \\ \langle x, y\rangle & 1 \end{bmatrix}. \tag{24}$$

We introduce Proposition 11.31 in O'Donnell [60]:

$$\mathop{\mathbb{E}}_{(a,b)\sim\mathcal{N}(0,\Sigma)} [h_\ell(a)\, h_m(b)] = \ell! \cdot \langle x, y\rangle^\ell \cdot \mathbb{1}(\ell = m). \tag{25}$$

This completes the proof of Proposition 1. □

B.2 Proof of Theorem 1

Theorem 1. *For a polynomial $\widetilde{\sigma}(t) = \sum_{j=0}^{q} a_j t^j$, the dual kernel of $\widetilde{\sigma}(\cdot)$, as per Definition 1, is*

$$K_{\widetilde{\sigma}}(x, y) := \sum_{\ell=0}^{q} r_{\widetilde{\sigma},\ell}(\|x\|_2)\, r_{\widetilde{\sigma},\ell}(\|y\|_2) \left(\frac{\langle x, y\rangle}{\|x\|_2 \|y\|_2}\right)^\ell \tag{9}$$

where $r_{\widetilde{\sigma},\ell}(t) := \sum_{i=0}^{\lfloor \frac{q-\ell}{2} \rfloor} \frac{a_{\ell+2i}(\ell+2i)!}{2^i \cdot i! \cdot \sqrt{\ell!}} t^{2i+\ell}$. Moreover, if an activation function $\sigma : \mathbb{R} \to \mathbb{R}$ satisfies $\|\sigma\|^2_{\mathcal{N}(0,\nu^2)} < \infty$ and $\|\sigma - \widetilde{\sigma}\|^2_{\mathcal{N}(0,\nu^2)} \le \varepsilon$ for some $\varepsilon > 0$ and $\nu \ge 1$, then for every $x, y \in \mathbb{R}^d$ such that $\|x\|_2, \|y\|_2 \in (0, \nu]$ the following holds

$$|K_\sigma(x, y) - K_{\widetilde{\sigma}}(x, y)| \le \sqrt{\frac{\nu^2 \cdot \varepsilon \left(6\, \|\sigma\|^2_{\mathcal{N}(0,\nu^2)} + 4\varepsilon\right)}{\|x\|_2 \|y\|_2}}. \tag{10}$$

Proof of Theorem 1: Due to homogeneity of the inner-product, we can write

$$\widetilde{\sigma}(\langle w, x\rangle) = \sum_{i=0}^{q} a_i \langle w, x\rangle^i = \sum_{i=0}^{q} a_i \|x\|_2^i \left\langle w, \frac{x}{\|x\|_2}\right\rangle^i. \tag{26}$$

Note that monomial t^i of degree $i \ge 0$ can be explicitly written in the Hermite basis as $t^i = \sum_{\ell=0}^{i} \mu_{i,\ell} h_\ell(t)$ where

$$\mu_{i,\ell} = \begin{cases} \dfrac{i!}{2^{\frac{i-\ell}{2}} \cdot (\frac{i-\ell}{2})! \cdot \ell!} & \text{if } i - \ell \text{ is even,} \\ 0 & \text{if } i - \ell \text{ is odd.} \end{cases} \tag{27}$$

Plugging this into Equation (26) and re-arranging terms, we obtain that

$$\widetilde{\sigma}(\langle w, x\rangle) = \sum_{\ell=0}^{q} \left(\sum_{i=\ell}^{q} a_i\, \mu_{i,\ell} \|x\|_2^i\right) h_\ell\left(\left\langle w, \frac{x}{\|x\|_2}\right\rangle\right). \tag{28}$$

Applying Equation (28) to the definition of the dual kernel $K_{\widetilde{\sigma}}(x, y)$ given in Equation (5) and taking the expectation over w gives

$$\begin{aligned} K_{\widetilde{\sigma}}(x, y) &= \mathop{\mathbb{E}}_{w\sim\mathcal{N}(0,\boldsymbol{I}_d)} [\widetilde{\sigma}(\langle w, x\rangle) \cdot \widetilde{\sigma}(\langle w, y\rangle)] \\ &= \mathop{\mathbb{E}}_{w} \left[\sum_{\ell=0}^{q} \sum_{m=0}^{q} \left(\sum_{i=\ell}^{q} a_i\, \mu_{i,\ell} \|x\|_2^i\right) \left(\sum_{j=m}^{q} a_j\, \mu_{j,m} \|y\|_2^j\right) h_\ell\left(\frac{\langle w, x\rangle}{\|x\|_2}\right) h_m\left(\frac{\langle w, y\rangle}{\|y\|_2}\right)\right] \\ &= \sum_{\ell=0}^{q} \sum_{m=0}^{q} \left(\sum_{i=\ell}^{q} a_i\, \mu_{i,\ell} \|x\|_2^i\right) \left(\sum_{j=m}^{q} a_j\, \mu_{j,m} \|y\|_2^j\right) \mathop{\mathbb{E}}_{w} \left[h_\ell\left(\frac{\langle w, x\rangle}{\|x\|_2}\right) h_m\left(\frac{\langle w, y\rangle}{\|y\|_2}\right)\right] \\ &= \sum_{\ell=0}^{q} \left(\sum_{i=\ell}^{q} a_i\, \mu_{i,\ell} \|x\|_2^i\right) \left(\sum_{j=\ell}^{q} a_j\, \mu_{j,\ell} \|y\|_2^j\right) \ell! \left(\frac{\langle x, y\rangle}{\|x\|_2 \|y\|_2}\right)^\ell, \end{aligned} \tag{29}$$

where Equation (29) follows from Proposition 1. Now using Equation (27), we have

$$\sum_{i=\ell}^{q} a_i \cdot \mu_{i,\ell} \|x\|_2^i = \sum_{k=0}^{\lfloor \frac{q-\ell}{2} \rfloor} \frac{a_{\ell+2k} \cdot (\ell+2k)!}{2^k \cdot k! \cdot \ell!} \|x\|_2^{\ell+2k}.$$

Therefore, we obtain that

$$K_{\widetilde{\sigma}}(x,y) = \sum_{\ell=0}^{q} \left(\sum_{i=0}^{\lfloor \frac{q-\ell}{2} \rfloor} \frac{a_{\ell+2i}(\ell+2i)!}{2^i \cdot i! \sqrt{\ell!}} \|x\|_2^{2i+\ell} \right) \left(\sum_{j=0}^{\lfloor \frac{q-\ell}{2} \rfloor} \frac{a_{\ell+2j}(\ell+2j)!}{2^j \cdot j! \sqrt{\ell!}} \|y\|_2^{2j+\ell} \right) \left(\frac{\langle x, y \rangle}{\|x\|_2 \|y\|_2} \right)^{\ell}.$$

We finish off the proof of Theorem 1 by bounding the error $|K_\sigma(x,y) - K_{\widetilde{\sigma}}(x,y)|$. We use Equation (5) along with the assumption that both $x, y \neq 0$ to write,

$$\begin{aligned}
|K_\sigma(x,y) - K_{\widetilde{\sigma}}(x,y)| &= \left| \mathop{\mathbb{E}}_{w} \left[\sigma(\langle w, x \rangle) \cdot \sigma(\langle w, y \rangle) - \widetilde{\sigma}(\langle w, x \rangle) \cdot \widetilde{\sigma}(\langle w, y \rangle) \right] \right| \\
&\leq \left| \mathop{\mathbb{E}}_{w} \left[(\sigma(\langle w, x \rangle) - \widetilde{\sigma}(\langle w, x \rangle)) \cdot \sigma(\langle w, y \rangle) \right] \right| && (30) \\
&\quad + \left| \mathop{\mathbb{E}}_{w} \left[(\sigma(\langle w, y \rangle) - \widetilde{\sigma}(\langle w, y \rangle)) \cdot \widetilde{\sigma}(\langle w, x \rangle) \right] \right| && (31)
\end{aligned}$$

where the inequality above follows from the triangle inequality. Now we bound each of Equation (30) and Equation (31) separately. First let us bound Equation (30) using Cauchy–Schwarz inequality as follows,

$$\begin{aligned}
\left| \mathop{\mathbb{E}}_{w} \left[(\sigma(\langle w, x \rangle) - \widetilde{\sigma}(\langle w, x \rangle)) \cdot \sigma(\langle w, y \rangle) \right] \right| &\leq \sqrt{\mathop{\mathbb{E}}_{w} \left[|\sigma(\langle w, x \rangle) - \widetilde{\sigma}(\langle w, x \rangle)|^2 \right] \cdot \mathop{\mathbb{E}}_{w} \left[\sigma(\langle w, y \rangle)^2 \right]} \\
&= \sqrt{\mathop{\mathbb{E}}_{\alpha \sim \mathcal{N}(0, \|x\|_2^2)} \left[|\sigma(\alpha) - \widetilde{\sigma}(\alpha)|^2 \right] \cdot \mathop{\mathbb{E}}_{\beta \sim \mathcal{N}(0, \|y\|_2^2)} \left[\sigma(\beta)^2 \right]} \\
&\leq \sqrt{\frac{\mathbb{E}_{\alpha \sim \mathcal{N}(0, \nu^2)} \left[|\sigma(\alpha) - \widetilde{\sigma}(\alpha)|^2 \right]}{\|x\|_2 / \nu} \cdot \mathop{\mathbb{E}}_{\beta \sim \mathcal{N}(0, \|y\|_2^2)} \left[\sigma(\beta)^2 \right]} \\
&\leq \sqrt{\frac{\varepsilon \cdot \nu}{\|x\|_2} \cdot \frac{\nu}{\|y\|_2} \mathop{\mathbb{E}}_{\beta \sim \mathcal{N}(0, \nu^2)} \left[\sigma(\beta)^2 \right]}, && (32)
\end{aligned}$$

where the first line follows from the Cauchy–Schwarz inequality and the third line above follows from the assumption that $\|x\|_2, \|y\|_2 \neq 0$ and the last line follows from the precondition of Theorem 1.

Similarly, we can bound Equation (31), as follows,

$$\begin{aligned}
&\left| \mathop{\mathbb{E}}_{w} \left[(\sigma(\langle w, y \rangle) - \widetilde{\sigma}(\langle w, y \rangle)) \cdot \widetilde{\sigma}(\langle w, x \rangle) \right] \right| \\
&\leq \sqrt{\mathop{\mathbb{E}}_{w} \left[|(\sigma(\langle w, y \rangle) - \widetilde{\sigma}(\langle w, y \rangle))|^2 \right] \cdot \mathop{\mathbb{E}}_{w} \left[|\widetilde{\sigma}(\langle w, x \rangle)|^2 \right]} \\
&= \sqrt{\mathop{\mathbb{E}}_{\alpha \sim \mathcal{N}(0, \|y\|_2^2)} \left[|\sigma(\alpha) - \widetilde{\sigma}(\alpha)|^2 \right] \cdot \mathop{\mathbb{E}}_{\beta \sim \mathcal{N}(0, \|x\|_2^2)} \left[|\widetilde{\sigma}(\beta)|^2 \right]} \\
&\leq \sqrt{\frac{\mathbb{E}_{\alpha \sim \mathcal{N}(0, \nu^2)} \left[|\sigma(\alpha) - \widetilde{\sigma}(\alpha)|^2 \right]}{\|y\|_2 / \nu} \cdot \mathop{\mathbb{E}}_{\beta \sim \mathcal{N}(0, \|x\|_2^2)} \left[|\widetilde{\sigma}(\beta)|^2 \right]} \\
&\leq \sqrt{\frac{\varepsilon \cdot \nu}{\|y\|_2} \cdot \mathop{\mathbb{E}}_{\beta \sim \mathcal{N}(0, \|x\|_2^2)} \left[2 |\sigma(\beta)|^2 + 2 |\widetilde{\sigma}(\beta) - \sigma(\beta)|^2 \right]} \\
&\leq \sqrt{\frac{\varepsilon \cdot \nu}{\|y\|_2} \cdot \frac{\nu}{\|x\|_2} \mathop{\mathbb{E}}_{\beta \sim \mathcal{N}(0, \nu^2)} \left[2 |\sigma(\beta)|^2 + 2 |\widetilde{\sigma}(\beta) - \sigma(\beta)|^2 \right]} \\
&\leq \sqrt{\frac{\varepsilon \cdot \nu}{\|y\|_2} \cdot \frac{2\nu(\varepsilon + \mathbb{E}_{\beta \sim \mathcal{N}(0, \nu^2)} [|\sigma(\beta)|^2])}{\|x\|_2}}, && (33)
\end{aligned}$$

where the fourth line above follows from the assumption that $\|x\|_2, \|y\|_2 \neq 0$, the fifth line above follows from the AM-GM inequality along with the the preconditions of Theorem 1, and the last equality above follows from the preconditions of Theorem 1.

Now by plugging Equation (32) and Equation (33) back into Equation (30) and Equation (31) we find that,

$$\begin{aligned}
|K_\sigma(x,y) - K_{\widetilde{\sigma}}(x,y)| &\leq \sqrt{\frac{\varepsilon \cdot \nu^2 \cdot \mathbb{E}_{\beta\sim\mathcal{N}(0,\nu^2)}\left[\sigma(\beta)^2\right]}{\|x\|_2 \cdot \|y\|_2}} \left(1 + \sqrt{2 + \frac{2\varepsilon}{\mathbb{E}_{\beta\sim\mathcal{N}(0,\nu^2)}\left[\sigma(\beta)^2\right]}}\right) \\
&\leq \sqrt{\frac{\varepsilon \cdot \nu^2 \cdot \mathbb{E}_{\beta\sim\mathcal{N}(0,\nu^2)}\left[\sigma(\beta)^2\right]}{\|x\|_2 \cdot \|y\|_2}} \cdot \sqrt{6 + \frac{4\varepsilon}{\mathbb{E}_{\beta\sim\mathcal{N}(0,\nu^2)}\left[\sigma(\beta)^2\right]}} \\
&= \sqrt{\frac{\varepsilon \cdot \nu^2}{\|x\|_2 \cdot \|y\|_2} \cdot \left(6 \mathop{\mathbb{E}}_{\beta\sim\mathcal{N}(0,\nu^2)}\left[\sigma(\beta)^2\right] + 4\varepsilon\right)}.
\end{aligned}$$

This completes the proof of Theorem 1. □

Examples for Taylor expansion. Observe that $\sigma(t) = \sin(t)$ is analytic and has a Taylor expansion with coefficients $a_{\ell+2i} = \frac{(-1)^{\frac{\ell+2i-1}{2}}}{(\ell+2i)!} \cdot \mathbb{1}(\ell \text{ is odd})$. By invoking Theorem 1 we have,

$$r_{\sigma,\ell}(t) = \mathbb{1}(\ell \text{ is odd}) \cdot \sum_{i=0}^{\infty} \frac{(-1)^{\frac{\ell+2i-1}{2}}}{(\ell+2i)!} \frac{(\ell+2i)!}{2^i \cdot i! \cdot \sqrt{\ell!}} t^{2i+\ell} = \mathbb{1}(\ell \text{ is odd}) \cdot \frac{(-1)^{\frac{\ell-1}{2}} t^\ell}{\sqrt{\ell!}} \cdot e^{-\frac{t^2}{2}}. \quad (34)$$

Therefore,

$$K_{\sin}(x,y) = \sum_{\ell=0}^{\infty} e^{-\frac{\|x\|_2^2}{2}} \cdot e^{-\frac{\|y\|_2^2}{2}} \cdot \frac{\langle x,y\rangle^{2\ell+1}}{(2\ell+1)!} = e^{-\frac{\|x\|_2^2+\|y\|_2^2}{2}} \sinh(\langle x,y\rangle). \quad (35)$$

Similarly, we can derive that $K_{\cos}(x,y) = e^{-\frac{\|x\|_2^2+\|y\|_2^2}{2}} \cosh(\langle x,y\rangle)$ that corresponds to Table 1.

B.3 Proof of Theorem 2

Theorem 2. *Given $\sigma : \mathbb{R} \to \mathbb{R}$, suppose that there exists an integer $k \geq 2$ and some $\nu \geq 1$ such that for every $i = 0, \ldots, k$, $\sigma^{(i)}$ is absolutely continuous and $\lim_{t\to\pm\infty} e^{-\frac{t^2}{4}} \sigma^{(i)}(\nu t) = 0$ and moreover $\|\sigma\|^2_{\mathcal{N}(0,\nu^2)} < \infty$ and $\left\|\sigma^{(k)}\right\|^2_{\mathcal{N}(0,\nu^2)} < \infty$. Consider the Hermite expansion coefficients $\{c_j\}_{j\geq 0}$ of function $\sigma(\nu t)$ and denote $\widetilde{\sigma}(t) := \sum_{j=0}^{q} c_j h_j(t/\nu)$. Given $x, y \in \mathbb{R}^d$ with $\|x\|_2, \|y\|_2 \in (0, \nu]$,*

$$|K_\sigma(x,y) - K_{\widetilde{\sigma}}(x,y)| \leq \frac{5\nu^{k+1} \left\|\sigma^{(k)}\right\|_{\mathcal{N}(0,\nu^2)} \max\left(\|\sigma\|_{\mathcal{N}(0,\nu^2)}, \nu^k \left\|\sigma^{(k)}\right\|_{\mathcal{N}(0,\nu^2)}\right)}{\sqrt{\|x\|_2 \|y\|_2 \cdot k \cdot q^{k-1}}}. \quad (11)$$

where $K_\sigma(\cdot,\cdot)$ and $K_{\widetilde{\sigma}}(\cdot,\cdot)$ are dual kernels corresponding to $\sigma(\cdot)$ and $\widetilde{\sigma}(\cdot)$ in Definition 1, respectively. Moreover, for the ReLU activation $\sigma(t) = \max(t, 0)$, it holds that

$$|K_\sigma(x,y) - K_{\widetilde{\sigma}}(x,y)| \leq \sqrt{\frac{2\nu^6}{q \|x\|_2 \|y\|_2}}. \quad (12)$$

In order to prove this theorem we first need to establish the following bound on the decay rate of the Hermite expansion coefficients of smooth functions,

Lemma 1. *Suppose that there exists an integer $k \geq 0$ such that for every $i = 0, \ldots, k$, $\sigma^{(i)}(t)$ are absolutely continuous in $\mathbb{R}$ and $\lim_{t\to\pm\infty} e^{-\frac{t^2}{4}} \sigma^{(i)}(t) = 0$. Assume that $\|\sigma\|^2_{\mathcal{N}} < \infty$ and $\left\|\sigma^{(k)}\right\|^2_{\mathcal{N}} < \infty$. Let $\{c_j\}_{j=0}^{\infty}$ be the Hermite expansion coefficients of this function such that $\left\|\sigma - \sum_{j=0}^{\infty} c_j h_j\right\|_{\mathcal{N}} = 0$. Then, for any integer $j \geq k$:*

$$|c_j| \leq \left\|\sigma^{(k)}\right\|_{\mathcal{N}} \frac{\sqrt{(j-k)!}}{j!}. \quad (36)$$

The proof of Lemma 1 is provided in Appendix B.4.

Proof of Theorem 2: First, because of the precondition of Theorem 2 about $\|\sigma\|^2_{\mathcal{N}(0,\nu^2)} = \mathbb{E}_{t\sim\mathcal{N}(0,\nu^2)}\left[|\sigma(t)|^2\right] = \mathbb{E}_{t\sim\mathcal{N}(0,1)}\left[|\sigma(\nu t)|^2\right] < \infty$, the function $\sigma(\nu t)$ is an L^2 function with respect to the normal measure $\mathcal{N}(0,1)$ on the real line. Therefore, because the Hermite polynomials $\{h_j\}_{j=0}^{\infty}$ provide an orthogonal basis for L^2 function with respect to normal measure $\mathcal{N}(0,1)$, $\sigma(\nu t)$ converges to its Hermite expansion, i.e., $\mathbb{E}_{t\sim\mathcal{N}(0,1)}\left[\left|\sigma(\nu t) - \sum_{j=0}^{\infty} c_j h_j(t)\right|^2\right] = 0$. We obtain an error bound on the dual kernel by invoking Theorem 1. To do so, we need to first upper bound $\mathbb{E}_{t\sim\mathcal{N}(0,\nu^2)}\left[|\sigma(t) - \widetilde{\sigma}(t)|^2\right]$, as follows

$$\begin{aligned}
\mathop{\mathbb{E}}_{t\sim\mathcal{N}(0,\nu^2)}\left[|\sigma(t) - \widetilde{\sigma}(t)|^2\right] &= \mathop{\mathbb{E}}_{t\sim\mathcal{N}(0,1)}\left[|\sigma(\nu t) - \widetilde{\sigma}(\nu t)|^2\right] \\
&= \sum_{j=q+1}^{\infty} |c_j|^2 \cdot \mathop{\mathbb{E}}_{t\sim\mathcal{N}(0,1)}\left[|h_j(t)|^2\right] \\
&= \sum_{j=q+1}^{\infty} |c_j|^2 \cdot j!, \qquad (37)
\end{aligned}$$

where the second line above follows from the fact that h_j's are orthogonal with respect to the normal measure $\mathcal{N}(0,1)$. The third line follows from the fact that $\|h_j\|^2_{\mathcal{N}} = j!$.

We now proceed to upper bound the term in Equation (37), using the bound on the Hermite expansion coefficients we proved in Lemma 1. We apply this lemma to the function $\sigma(\nu t)$ whose Hermite expansion coefficients are $\{c_i\}_{i=0}^{\infty}$. By precondition of Theorem 2 we have $\left\|\sigma^{(k)}\right\|^2_{\mathcal{N}} = \mathbb{E}_{t\sim\mathcal{N}(0,\nu^2)}\left[\left|\sigma^{(k)}(t)\right|^2\right] < \infty$. This implies that,

$$\mathop{\mathbb{E}}_{t\sim\mathcal{N}(0,1)}\left[\left|\frac{d^k}{dt^k}\sigma(\nu t)\right|^2\right] = \nu^{2k} \cdot \mathop{\mathbb{E}}_{t\sim\mathcal{N}(0,1)}\left[\left|\frac{d^k}{d(\nu t)^k}\sigma(\nu t)\right|^2\right] = \nu^{2k} \cdot \left\|\sigma^{(k)}\right\|^2_{\mathcal{N}(0,\nu^2)} < \infty.$$

Furthermore, the precondition of Theorem 2 about $\lim_{t\to\pm\infty} e^{-\frac{t^2}{4}}\sigma^{(i)}(\nu t) = 0$ implies the following,

$$\lim_{t\to\pm\infty} e^{-\frac{t^2}{4}}\frac{d^i}{dt^i}\sigma(\nu t) = \nu^i \cdot \lim_{t\to\pm\infty} e^{-\frac{t^2}{4}}\frac{d^i}{d(\nu t)^i}\sigma(\nu t) = 0.$$

Therefore, the preconditions of Lemma 1 are satisfied and by invoking this lemma we have the following inequality for any integer $j \geq k$,

$$|c_j| \leq \sqrt{\mathop{\mathbb{E}}_{t\sim\mathcal{N}(0,1)}\left[\left|\frac{d^k}{dt^k}\sigma(\nu t)\right|^2\right]} \cdot \frac{\sqrt{(j-k)!}}{j!} = \nu^k \cdot \left\|\sigma^{(k)}\right\|_{\mathcal{N}(0,\nu^2)} \cdot \frac{\sqrt{(j-k)!}}{j!}.$$

Plugging the above inequality into Equation (36) into Equation (37), gives

$$\begin{aligned}
\mathop{\mathbb{E}}_{t\sim\mathcal{N}(0,\nu^2)}\left[|\sigma(t) - \widetilde{\sigma}(t)|^2\right] &= \sum_{j=q+1}^{\infty} |c_j|^2 \cdot j! \\
&\leq \nu^{2k} \cdot \left\|\sigma^{(k)}\right\|^2_{\mathcal{N}(0,\nu^2)} \cdot \sum_{j=q+1}^{\infty} \frac{(j-k)!}{j!} \\
&= \frac{\nu^{2k} \cdot \left\|\sigma^{(k)}\right\|^2_{\mathcal{N}(0,\nu^2)}}{k-1} \cdot \frac{1}{q(q-1)\cdots(q-k+2)} \\
&\leq \frac{\nu^{2k} \cdot \left\|\sigma^{(k)}\right\|^2_{\mathcal{N}(0,\nu^2)}}{k \cdot q^{k-1}} \qquad (38)
\end{aligned}$$

Thus we can now invoke Theorem 1 with $\varepsilon = \frac{\nu^{2k} \cdot \left\|\sigma^{(k)}\right\|^2_{\mathcal{N}(0,\nu^2)}}{k \cdot q^{k-1}}$ to find that

$$
\begin{aligned}
&|K_\sigma(x,y) - K_{\widetilde{\sigma}}(x,y)| \\
&\leq \sqrt{\frac{\varepsilon \cdot \nu^2}{\|x\|_2 \|y\|_2} \left(6 \|\sigma\|^2_{\mathcal{N}(0,\nu^2)} + 4\varepsilon\right)} \\
&\leq \sqrt{\frac{\nu^2}{\|x\|_2 \|y\|_2} \left(6 \|\sigma\|^2_{\mathcal{N}(0,\nu^2)} \frac{\nu^{2k} \cdot \left\|\sigma^{(k)}\right\|^2_{\mathcal{N}(0,\nu^2)}}{k \cdot q^{k-1}} + 4 \left(\frac{\nu^{2k} \cdot \left\|\sigma^{(k)}\right\|^2_{\mathcal{N}(0,\nu^2)}}{k \cdot q^{k-1}}\right)^2\right)} \\
&\leq \sqrt{\frac{\nu^2}{\|x\|_2 \|y\|_2}} \left(\sqrt{6 \|\sigma\|^2_{\mathcal{N}(0,\nu^2)} \frac{\nu^{2k} \cdot \left\|\sigma^{(k)}\right\|^2_{\mathcal{N}(0,\nu^2)}}{k \cdot q^{k-1}}} + \sqrt{4 \left(\frac{\nu^{2k} \cdot \left\|\sigma^{(k)}\right\|^2_{\mathcal{N}(0,\nu^2)}}{k \cdot q^{k-1}}\right)^2}\right) \\
&\leq \frac{\nu}{\sqrt{\|x\|_2 \|y\|_2}} \left(\|\sigma\|_{\mathcal{N}(0,\nu^2)} \nu^k \left\|\sigma^{(k)}\right\|_{\mathcal{N}(0,\nu^2)} \sqrt{\frac{6}{k \cdot q^{k-1}}} + 2 \left(\frac{\nu^{2k} \cdot \left\|\sigma^{(k)}\right\|^2_{\mathcal{N}(0,\nu^2)}}{k \cdot q^{k-1}}\right)\right) \\
&\leq \frac{\nu^{k+1} \left\|\sigma^{(k)}\right\|_{\mathcal{N}(0,\nu^2)}}{\sqrt{\|x\|_2 \|y\|_2}} \max\left(\|\sigma\|_{\mathcal{N}(0,\nu^2)}, \nu^k \left\|\sigma^{(k)}\right\|_{\mathcal{N}(0,\nu^2)}\right) \left(\sqrt{\frac{6}{k \cdot q^{k-1}}} + \frac{2}{k \cdot q^{k-1}}\right) \\
&\leq \frac{5\nu^{k+1} \left\|\sigma^{(k)}\right\|_{\mathcal{N}(0,\nu^2)} \max\left(\|\sigma\|_{\mathcal{N}(0,\nu^2)}, \nu^k \left\|\sigma^{(k)}\right\|_{\mathcal{N}(0,\nu^2)}\right)}{\sqrt{\|x\|_2 \|y\|_2 \cdot k \cdot q^{k-1}}}.
\end{aligned}
$$

Now we prove the second statement of the theorem about the ReLU activation $\sigma(t) = \max(t, 0)$. It is easy to check that for this function

$$
\|\sigma\|^2_{\mathcal{N}(0,\nu^2)} = \mathop{\mathbb{E}}_{t \sim \mathcal{N}(0,\nu^2)}[|\sigma(t)|^2] = \frac{\nu^2}{\sqrt{2\pi}} \int_0^\infty t^2 \cdot e^{-\frac{t^2}{2}} dt = \frac{\nu^2}{2}. \tag{39}
$$

Furthermore for any $j \geq 0$, the Hermite coefficients of $\sigma(\nu t)$ are

$$
c_j = \frac{1}{\sqrt{2\pi} j!} \int_{-\infty}^\infty \max(\nu t, 0) \cdot h_j(t) \cdot e^{-\frac{t^2}{2}} dt = \frac{\nu}{\sqrt{2\pi} j!} \int_0^\infty t \cdot h_j(t) \cdot e^{-\frac{t^2}{2}} dt
$$

Using integration-by-parts and the fact that $h_j'(t) = j h_{j-1}(t)$ for all $j \geq 1$, we get that

$$
\begin{aligned}
\int_0^\infty t \cdot h_j(t) \cdot e^{-\frac{t^2}{2}} dt &= h_j(x) \left(-e^{-\frac{t^2}{2}}\right)\Big|_0^\infty + \int_0^\infty h_j'(t) \cdot e^{-\frac{t^2}{2}} dt \\
&= h_j(0) + \int_0^\infty h_j'(t) \cdot e^{-\frac{t^2}{2}} dt \\
&= h_j(0) + j \int_0^\infty h_{j-1}(t) \cdot e^{-\frac{t^2}{2}} dt \\
&= (-1)^{\frac{j}{2}} \cdot (j-1)!! \cdot \mathbb{1}_{\{j \text{ is even}\}} + j \cdot (-1)^{\frac{j}{2}-1} \cdot (j-3)!! \cdot \mathbb{1}_{\{j \text{ is even}\}} \\
&= (-1)^{\frac{j}{2}-1} \cdot (j-3)!! \cdot \mathbb{1}_{\{j \text{ is even}\}}.
\end{aligned}
$$

Therefore,

$$
\mathop{\mathbb{E}}_{t \sim \mathcal{N}(0,\nu^2)} \left[|\sigma(t) - \widetilde{\sigma}(t)|^2\right] = \sum_{j=q+1}^\infty c_j^2 \cdot \sqrt{2\pi} j! = \sum_{j=q+1}^\infty \nu^2 \cdot \frac{((j-3)!!)^2}{\sqrt{2\pi} j!} = \frac{\nu^2}{\sqrt{2\pi}(q+1)}, \tag{40}
$$

By invoking Theorem 1, using Equation (39) and Equation (40), we have

$$
\begin{aligned}
|K_\sigma(x,y) - K_{\widetilde{\sigma}}(x,y)| &\leq \sqrt{\frac{\varepsilon \cdot \nu^2}{\|x\|_2 \|y\|_2}\left(6\|\sigma\|^2_{\mathcal{N}(0,\nu^2)} + 4\varepsilon\right)} \\
&= \sqrt{\frac{\frac{\nu^2}{\sqrt{2\pi}(q+1)} \cdot \nu^2}{\|x\|_2 \|y\|_2}\left(3\nu^2 + 4\frac{\nu^2}{\sqrt{2\pi}(q+1)}\right)} \\
&\leq \sqrt{\frac{2\nu^6}{(q+1)\|x\|_2 \|y\|_2}}
\end{aligned}
$$

This completes the proof of Theorem 2. □

B.4 Proof of Lemma 1

Lemma 1. *Suppose that there exists an integer $k \geq 0$ such that for every $i = 0, \ldots, k$, $\sigma^{(i)}(t)$ are absolutely continuous in $\mathbb{R}$ and $\lim_{t\to\pm\infty} e^{-\frac{t^2}{4}}\sigma^{(i)}(t) = 0$. Assume that $\|\sigma\|^2_{\mathcal{N}} < \infty$ and $\left\|\sigma^{(k)}\right\|^2_{\mathcal{N}} < \infty$. Let $\{c_j\}_{j=0}^{\infty}$ be the Hermite expansion coefficients of this function such that $\left\|\sigma - \sum_{j=0}^{\infty} c_j h_j\right\|_{\mathcal{N}} = 0$. Then, for any integer $j \geq k$:*

$$
|c_j| \leq \left\|\sigma^{(k)}\right\|_{\mathcal{N}} \frac{\sqrt{(j-k)!}}{j!}. \tag{36}
$$

Proof of Lemma 1: The proof can be obtained by slightly modifying Theorem 3.1 in [61]. The precondition $\|\sigma\|^2_{\mathcal{N}} = \mathbb{E}_{t\sim\mathcal{N}(0,1)}\left[|\sigma(t)|^2\right] < \infty$ implies that σ is an L^2-function with respect to measure $e^{-\frac{t^2}{2}}$ on real line. Because Hermite polynomials $\{h_j\}_{j=0}^{\infty}$ form an orthogonal basis for the Hilbert space of L^2-functions with respect to normal measure $\mathcal{N}(0,1)$, $\sigma(t)$ converges to its Hermite expansion, i.e., $\sum_{j=0}^{\infty} c_j h_j(t)$. The j-th coefficient in this expansion is

$$
c_j = \frac{1}{\sqrt{2\pi}j!}\int_{-\infty}^{\infty} \sigma(t) \cdot h_j(t) \cdot e^{-\frac{t^2}{2}}\, dt. \tag{41}
$$

Using the Rodrigues' expression of Hermite polynomials in Equation (1) and integration-by-parts, we have,

$$
\begin{aligned}
\int_{-\infty}^{\infty} \sigma(t) h_j(t) e^{-\frac{t^2}{2}}\, dt &= (-1)^j \int_{-\infty}^{\infty} \sigma(t)\left[\frac{d^j}{dt^j} e^{-\frac{t^2}{2}}\right] dt \\
&= (-1)^j \sigma(t)\left[\frac{d^{j-1}}{dt^{j-1}} e^{-\frac{t^2}{2}}\right]\Bigg|_{-\infty}^{\infty} + (-1)^{j-1}\int_{-\infty}^{\infty} \sigma^{(1)}(t)\left[\frac{d^{j-1}}{dt^{j-1}} e^{-\frac{t^2}{2}}\right] dt \\
&= -\sigma(t) \cdot h_{j-1}(t) \cdot e^{-\frac{t^2}{2}}\Bigg|_{-\infty}^{\infty} + \int_{-\infty}^{\infty} \sigma^{(1)}(t) h_{j-1}(t) e^{-\frac{t^2}{2}}\, dt,
\end{aligned} \tag{42}
$$

where the last line above follows from the Rodrigues' expression of degree $j-1$ Hermite polynomial in Equation (1). Therefore, using 22.14.17 in [55][3], the first term in Equation (42) is 0 and by

[3] Equation 22.14.17 in [55] was $|H_j(t)| \leq a_0 e^{\frac{t^2}{2}} 2^{\frac{j}{2}} \sqrt{j!}$ where $a_0 \approx 1.086435$ and $H_j(\cdot)$ is physicist's Hermite polynomial. Using $H_j(t) = 2^{\frac{j}{2}} h_j(\sqrt{2}t)$ gives that $|e^{-\frac{t^2}{4}} h_j(t)| \leq a_0\sqrt{j!}$.

applying the above repeatedly we have

$$\begin{aligned}\left|\int_{-\infty}^{\infty}\sigma(t)\cdot h_j(t)\cdot e^{-\frac{t^2}{2}}dt\right| &= \left|\int_{-\infty}^{\infty}\sigma^{(1)}(t)\cdot h_{j-1}(t)\cdot e^{-\frac{t^2}{2}}dt\right| \\ &\vdots \\ &= \left|\int_{-\infty}^{\infty}\sigma^{(k)}(t)\cdot h_{j-k}(t)\cdot e^{-\frac{t^2}{2}}dt\right| \\ &\leq \sqrt{\int_{-\infty}^{\infty}\left|\sigma^{(k)}(t)\right|^2 e^{-\frac{t^2}{2}}dt\cdot\int_{-\infty}^{\infty}\left|h_{j-k}(t)\right|^2 e^{-\frac{t^2}{2}}dt} \\ &= \sqrt{2\pi}\cdot\left\|\sigma^{(k)}\right\|_{\mathcal{N}}\cdot\sqrt{(j-k)!} \end{aligned} \tag{43}$$

where the second last inequality comes from Cauchy-Schwarz inequality and the last one holds from that $\mathbb{E}_{t\sim\mathcal{N}(0,1)}\left[\left|h_\ell(t)\right|^2\right]=\ell!$ and the assumption. This completes the proof of Lemma 1. □

B.5 Proof of Theorem 3

Theorem 3. *Given a differentiable activation function $\sigma:\mathbb{R}\to\mathbb{R}$ which satisfies $|\sigma(t)|\leq C_1\exp\left(\frac{t^2}{4.1\nu^2}\right)$, $|\sigma'(t)|\leq C_2\exp\left(\frac{t^2}{4.1\nu^2}\right)$, $\|\sigma\|^2_{\mathcal{N}(0,\nu^2)}<\infty$ and $\|\sigma''\|^2_{\mathcal{N}(0,\nu^2)}<\infty$ for some $\nu\geq 1$ and constants C_1,C_2, the following holds for any $x,y\in\mathbb{R}^d$ with $\|x\|_2,\|y\|_2\in(0,\nu]$ and $|\langle x,y\rangle|<\|x\|_2\|y\|_2$:*

$$K_{\sigma'}(x,y)=\frac{1}{\|x\|_2\|y\|_2}\frac{\partial}{\partial c}k_\sigma\left(\|x\|_2,\|y\|_2,c\right)\Bigg|_{c=\frac{\langle x,y\rangle}{\|x\|_2\|y\|_2}}. \tag{14}$$

Additionally, if $\frac{\partial}{\partial c}k_\sigma(\cdot,\cdot,c)$ is continuous at $c=\pm1$ then Equation (14) holds for x,y such that $|\langle x,y\rangle|=\|x\|_2\|y\|_2$.

Note that our assumption on σ can be weakened to be: there exists $\epsilon>0$ and constants C_1 and C_2 such that

$$|\sigma''(t)|\leq C_1e^{C_2|t|^{2-\epsilon}}. \tag{44}$$

Proof of Theorem 3: Recall that the dual activation is defined as

$$k_\sigma(a,b,c):=\mathop{\mathbb{E}}_{(u,v)\sim\mathcal{N}(0,\mathbf{\Lambda})}\left[\sigma(u)\sigma(v)\right]. \tag{45}$$

where for $a,b\in\mathbb{R}_{\geq0}$ and $c\in[-1,1]$

$$\mathbf{\Lambda}:=\begin{bmatrix}a^2 & abc\\ abc & b^2\end{bmatrix}=\begin{bmatrix}a & 0\\ bc & b\sqrt{1-c^2}\end{bmatrix}\begin{bmatrix}a & 0\\ bc & b\sqrt{1-c^2}\end{bmatrix}^\top$$

Using a whitening transformation, we introduce the standard i.i.d. Gaussian random variables $w_1,w_2\sim\mathcal{N}(0,1)$ that satisfy

$$\begin{bmatrix}u\\ v\end{bmatrix}=\begin{bmatrix}a & 0\\ bc & b\sqrt{1-c^2}\end{bmatrix}\begin{bmatrix}w_1\\ w_2\end{bmatrix}.$$

Thus, by denoting $w=\begin{bmatrix}w_1\\ w_2\end{bmatrix}$, the dual activation can be written as

$$k_\sigma(a,b,c)=\mathop{\mathbb{E}}_{w\sim\mathcal{N}(0,\boldsymbol{I}_2)}\left[\sigma(aw_1)\cdot\sigma(bcw_1+b\sqrt{1-c^2}w_2)\right]. \tag{46}$$

Using Equation (46), we can calculate $\frac{\partial}{\partial c}k_\sigma(\cdot,\cdot,c)$ if the derivative can be interchangeable with the expectation. To this end, we use the "measure theory" statement of *Leibniz integral rule*.

Lemma 2 (Measure theory statement of Leibniz integral rule, Theorem 6.28 of [62]). *Let μ be a probability distribution with support Ω, let $I \subset \mathbb{R}$ be a nontrivial open interval, also let $f : \Omega \times I \to \mathbb{R}$ be a map with the following properties:*

1. *For any $x \in I$, $\mathbb{E}_{w\sim\mu}[|f(w,x)|] < \infty$.*
2. *For almost all $w \in \Omega$, the map $x \to f(w,x)$ is differentiable with derivative $\frac{\partial}{\partial x} f(w,x)$.*
3. *There is a map $h : \Omega \to \mathbb{R}$ such that $\mathbb{E}_{w\sim\mu}[|h(w)|] < \infty$ and $\left|\frac{\partial}{\partial x} f(\cdot, x)\right| \leq h$.*

Then, for any $x \in I$, $\mathbb{E}_{w\sim\mu}\left[\left|\frac{\partial}{\partial x} f(w,x)\right|\right] < \infty$ and the function $F : x \to \mathbb{E}_{w\sim\mu}[f(w,x)]$ is differentiable with derivative $F'(x) = \mathbb{E}_{w\sim\mu}\left[\frac{\partial}{\partial x} f(w,x)\right]$.

To invoke Lemma 2 on the above expression of the dual kernel, we let I be the open interval $(-1+\varepsilon, 1-\varepsilon)$ for an arbitrarily small $\varepsilon > 0$ and $f : \mathbb{R}^2 \times I \to \mathbb{R}$ be the function defined as $f(w,c) := \sigma(aw_1) \cdot \sigma(bcw_1 + b\sqrt{1-c^2}w_2)$ for some fixed values of a, b. With these notations in place, we proceed to check if the preconditions of Lemma 2 are satisfied. To verify the first precondition, we need to show that for any $c \in I$, $\mathbb{E}_{w\sim\mathcal{N}(0,\boldsymbol{I}_2)}\left[\left|\sigma(aw_1) \cdot \sigma(bcw_1 + b\sqrt{1-c^2}w_2)\right|\right] < \infty$. We find that,

$$\begin{aligned}\mathop{\mathbb{E}}_{w\sim\mathcal{N}(0,\boldsymbol{I}_2)}\left[\left|\sigma(aw_1) \cdot \sigma(bcw_1 + b\sqrt{1-c^2}w_2)\right|\right] &\leq \sqrt{\mathop{\mathbb{E}}_{w_1}\left[|\sigma(aw_1)|^2\right] \mathop{\mathbb{E}}_{w}\left[\left|\sigma\left(bcw_1 + b\sqrt{1-c^2}w_2\right)\right|^2\right]} \\ &= \sqrt{\mathop{\mathbb{E}}_{w_1}\left[|\sigma(aw_1)|^2\right] \cdot \mathop{\mathbb{E}}_{\gamma\sim\mathcal{N}(0,1)}\left[|\sigma(b\gamma)|^2\right]},\end{aligned} \tag{47}$$

where the first line above follows from Cauchy–Schwarz inequality. The second line above follows from the fact that w_1 and w_2 are independent copies of the normal random variable $\mathcal{N}(0,1)$, thus the random variable $bcw_1 + b\sqrt{1-c^2}w_2$ is indeed $b\gamma$ for a normal $\gamma \sim \mathcal{N}(0,1)$. Now using the preconditions of Theorem 3, for any $a, b \in (0, \nu]$ we have

$$\mathop{\mathbb{E}}_{w_1\sim\mathcal{N}(0,1)}\left[|\sigma(aw_1)|^2\right] \leq \frac{\nu}{a}\|\sigma\|^2_{\mathcal{N}(0,\nu^2)} < \infty, \text{ and } \mathop{\mathbb{E}}_{\gamma\sim\mathcal{N}(0,1)}\left[|\sigma(b\gamma)|^2\right] \leq \frac{\nu}{b}\|\sigma\|^2_{\mathcal{N}(0,\nu^2)} < \infty. \tag{48}$$

Also in case $a = 0$ or $b = 0$ we have $\mathbb{E}_{w_1\sim\mathcal{N}(0,1)}\left[|\sigma(aw_1)|^2\right] = |\sigma(0)|^2 < \infty$, therefore, above inequalities along with Equation (47) proves the first precondition of Lemma 2 for any $a, b \in [0, \nu]$.

To verify that the second precondition of Lemma 2 holds, we show that for almost all $w_1, w_2 \in \mathbb{R}$ the map $c \to \sigma(aw_1) \cdot \sigma(bcw_1 + b\sqrt{1-c^2}w_2)$ is differentiable. This holds true because of the assumption of Theorem 3 on about the activation $\sigma(\cdot)$ being differentiable. The derivative of this map is in fact

$$\sigma(aw_1) \cdot \sigma'(bcw_1 + b\sqrt{1-c^2}w_2) \cdot \left(bw_1 - \frac{bc}{\sqrt{1-c^2}}w_2\right).$$

Finally, we check the third precondition of Lemma 2. Since $|c| < 1$, there is an $\varepsilon > 0$ such that $c \in (-1+\varepsilon, 1-\varepsilon)$. We have,

$$\begin{aligned}&\left|\sigma(aw_1) \cdot \sigma'(bcw_1 + b\sqrt{1-c^2}w_2) \cdot \left(bw_1 - \frac{bc}{\sqrt{1-c^2}}w_2\right)\right| \\ &\leq \left|\sigma(aw_1) \cdot \sigma'(bcw_1 + b\sqrt{1-c^2}w_2)\right| \cdot \left(b|w_1| + \frac{b}{\varepsilon}|w_2|\right) \\ &\leq C_1C_2 \exp\left(\frac{a^2w_1^2 + (bcw_1 + b\sqrt{1-c^2}w_2)^2}{4.1\nu^2}\right) \cdot \left(b|w_1| + \frac{b}{\varepsilon}|w_2|\right) \\ &\leq C_1C_2 \exp\left(\frac{(a^2+b^2)w_1^2 + b^2w_2^2}{4.1\nu^2}\right) \cdot \left(b|w_1| + \frac{b}{\varepsilon}|w_2|\right) \\ &\leq C_1C_2 \exp\left(\frac{w_1^2 + w_2^2}{2.05}\right) \cdot \left(b|w_1| + \frac{b}{\varepsilon}|w_2|\right) =: h(w),\end{aligned} \tag{49}$$

where the second inequality follows from the preconditions of Theorem 3 about the upper bounds on $\sigma(\cdot)$ and $\sigma'(\cdot)$, the third one follows from $(cw_1+\sqrt{1-c^2}w_2)^2 \leq w_1^2+w_2^2$, and the fourth one follows from $a, b \leq \nu$. Now it is easy to check that this upper bound function satisfies $\mathbb{E}_{w\sim\mathcal{N}(0,\boldsymbol{I}_2)}[|h(w)|] < \infty$.

Therefore, we can invoke Lemma 2 to calculate the derivative of the dual kernel $k_\sigma(a,b,c)$ with respect to c as follows,

$$\begin{aligned}
\frac{\partial}{\partial c} k_\sigma(a,b,c) &= \frac{\partial}{\partial c} \mathop{\mathbb{E}}_{w\sim\mathcal{N}(0,\boldsymbol{I}_2)} \left[\sigma(aw_1)\cdot\sigma(bcw_1+b\sqrt{1-c^2}w_2)\right] \\
&= \mathop{\mathbb{E}}_{w} \left[\frac{\partial}{\partial c}\left(\sigma(aw_1)\cdot\sigma(bcw_1+b\sqrt{1-c^2}w_2)\right)\right] \\
&= \mathop{\mathbb{E}}_{w} \left[\sigma(aw_1)\cdot\sigma'\left(bcw_1+b\sqrt{1-c^2}w_2\right)\cdot bw_1\right] \qquad (50) \\
&\qquad - \mathop{\mathbb{E}}_{w} \left[\sigma(aw_1)\cdot\sigma'\left(bcw_1+b\sqrt{1-c^2}w_2\right)\frac{bc}{\sqrt{1-c^2}}w_2\right]. \qquad (51)
\end{aligned}$$

Next we compute Equations (50) and (51) by using Stein's lemma,

Lemma 3 (Stein's Lemma)**.** *For a differentiable function $\phi:\mathbb{R}\to\mathbb{R}$ with $\mathbb{E}_{x\sim\mathcal{N}(0,1)}[|\phi'(x)|]<\infty$,*

$$\mathop{\mathbb{E}}_{x\sim\mathcal{N}(0,1)}[\phi(x)x] = \mathop{\mathbb{E}}_{x\sim\mathcal{N}(0,1)}[\phi'(x)]\ .$$

Applying Stein's Lemma to Equation (50) gives,

$$\begin{aligned}
&\mathop{\mathbb{E}}_{w} \left[\sigma(aw_1)\cdot\sigma'\left(bcw_1+b\sqrt{1-c^2}w_2\right)\cdot bw_1\right] \\
&= ab\mathop{\mathbb{E}}_{w} \left[\sigma'(aw_1)\cdot\sigma'\left(bcw_1+b\sqrt{1-c^2}w_2\right)\right] + b^2c\mathop{\mathbb{E}}_{w} \left[\sigma(aw_1)\cdot\sigma''\left(bcw_1+b\sqrt{1-c^2}w_2\right)\right].
\end{aligned} \tag{52}$$

Applying Stein's Lemma to Equation (51) gives,

$$\begin{aligned}
&-\mathop{\mathbb{E}}_{w} \left[\sigma(aw_1)\cdot\sigma'\left(bcw_1+b\sqrt{1-c^2}w_2\right)\frac{bc}{\sqrt{1-c^2}}w_2\right] \\
&= -b^2c\mathop{\mathbb{E}}_{w} \left[\sigma(aw_1)\cdot\sigma''\left(bcw_1+b\sqrt{1-c^2}w_2\right)\right]. \qquad (53)
\end{aligned}$$

Here we show that the term $b^2c\,\mathbb{E}_w\left[\sigma(aw_1)\cdot\sigma''\left(bcw_1+b\sqrt{1-c^2}w_2\right)\right]$ in Equations (52) and (53) has a bounded value as follows,

$$\begin{aligned}
\left|b^2c\mathop{\mathbb{E}}_{w} \left[\sigma(aw_1)\cdot\sigma''\left(bcw_1+b\sqrt{1-c^2}w_2\right)\right]\right| &\leq |b^2c|\cdot\sqrt{\mathop{\mathbb{E}}_{w_1}\left[|\sigma(aw_1)|^2\right]\mathop{\mathbb{E}}_{w}\left[\left|\sigma''\left(bcw_1+b\sqrt{1-c^2}w_2\right)\right|^2\right]} \\
&= |b^2c|\cdot\sqrt{\mathop{\mathbb{E}}_{w_1}\left[|\sigma(aw_1)|^2\right]\cdot\mathop{\mathbb{E}}_{\gamma\sim\mathcal{N}(0,1)}\left[|\sigma''(b\gamma)|^2\right]},
\end{aligned}$$

where the first line above follows from Cauchy–Schwarz inequality. The second line above follows from the fact that w_1 and w_2 are independent copies of the normal random variable $\mathcal{N}(0,1)$, thus the random variable $bcw_1+b\sqrt{1-c^2}w_2$ is indeed $b\gamma$ for a normal $\gamma\sim\mathcal{N}(0,1)$. Therefore, in order for the expectation $b^2c\,\mathbb{E}_w\left[\sigma(aw_1)\cdot\sigma''\left(bcw_1+b\sqrt{1-c^2}w_2\right)\right]$ to make sense, it is enough to have $b^2\,\mathbb{E}_{w_1}\left[|\sigma(aw_1)|^2\right]<\infty$ and $b^2\,\mathbb{E}_\gamma\left[|\sigma''(b\gamma)|^2\right]<\infty$. Note that the dual activation is symmetric with respect to swapping a and b, in the sense that $k_\sigma(a,b,c)=k_\sigma(b,a,c)$. Thus, we can without loss of generality assume that $b\leq a$. With this assumption $b^2\,\mathbb{E}_{w_1}\left[|\sigma(aw_1)|^2\right]\leq a^2\,\mathbb{E}_{w_1}\left[|\sigma(aw_1)|^2\right]$. Now, by recalling Equation (48), we have $a^2\,\mathbb{E}_{w_1}\left[|\sigma(aw_1)|^2\right]\leq a\nu\cdot\|\sigma\|^2_{\mathcal{N}(0,\nu^2)}<\infty$ and $b^2\,\mathbb{E}_\gamma\left[|\sigma''(b\gamma)|^2\right]\leq b\nu\cdot\|\sigma''\|^2_{\mathcal{N}(0,\nu^2)}<\infty$.

Summing Equations (52) and (53) and dividing the sum by ab give that

$$\frac{1}{ab}\frac{\partial}{\partial c}k_\sigma(a,b,c) = \mathop{\mathbb{E}}_{w\sim\mathcal{N}(0,\boldsymbol{I}_2)}\left[\sigma'(aw_1)\cdot\sigma'\left(bcw_1 + b\sqrt{1-c^2}w_2\right)\right]. \tag{54}$$

Finally, plugging in the values $a = \|x\|_2, b = \|y\|_2$ and $c = \frac{\langle x,y\rangle}{\|x\|_2\|y\|_2}$ such that $a, b \le \nu$ and $|c| < 1$ and using Equation (6) result in Equation (14).

Now suppose that the map $c \to \frac{\partial}{\partial c}k_\sigma(\cdot,\cdot,c)$ is continuous at $c = \pm 1$. Since we consider that $\sigma''(\cdot)$ exists, $\sigma'(\cdot)$ is continuous almost everywhere. Using these properties, we claim that the right-hand side in Equation (54) is continuous in c because for every $c' \in [-1, 1]$ it holds that

$$\begin{aligned}&\lim_{c\to c'}\mathop{\mathbb{E}}_{w\sim\mathcal{N}(0,\boldsymbol{I}_2)}\left[\sigma'(aw_1)\cdot\sigma'\left(bcw_1 + b\sqrt{1-c^2}w_2\right)\right]\\ &= \mathop{\mathbb{E}}_{w\sim\mathcal{N}(0,\boldsymbol{I}_2)}\left[\sigma'(aw_1)\cdot\lim_{c\to c'}\sigma'\left(bcw_1 + b\sqrt{1-c^2}w_2\right)\right].\end{aligned} \tag{55}$$

The above equality holds from the dominated convergence theorem (see Corollary 6.26 in [62]) with the dominated function obtained as

$$\begin{aligned}\left|\sigma'(aw_1)\cdot\sigma'\left(bcw_1 + b\sqrt{1-c^2}w_2\right)\right| &\le C_1C_2\exp\left(\frac{a^2w_1^2 + \left(bcw_1 + b\sqrt{1-c^2}w_2\right)^2}{4.1\nu^2}\right)\\ &\le C_1C_2\exp\left(\frac{(a^2+b^2)w_1^2 + b^2w_2^2}{4.1\nu^2}\right)\\ &\le C_1C_2\exp\left(\frac{w_1^2+w_2^2}{2.05}\right) := h'(w)\end{aligned} \tag{56}$$

where the first inequality follows from the preconditions of Theorem 3 and the second one follows from $(cw_1 + \sqrt{1-c^2}w_2)^2 \le w_1^2 + w_2^2$, and the third one follows from $a, b \le \nu$. And it is easy to check that $\mathbb{E}_{w\sim\mathcal{N}(0,\boldsymbol{I}_2)}[|h'(w)|] < \infty$. Hence both sides of Equation (54) are continuous at $c = \pm 1$ and taking $\lim_{c\to\pm 1}$ in both sides of Equation (54) gives that Equation (14) holds for x, y such that $|\langle x, y\rangle| = \|x\|_2\|y\|_2$. This concludes the proof of Theorem 3. □

Examples. For $\sigma(t) = \sin(t)$, the corresponding dual kernel is known to be

$$k_{\sin}(a,b,c) = e^{-\frac{a^2+b^2}{2}}\,\frac{e^{abc} - e^{-abc}}{2}. \tag{57}$$

Applying Theorem 3 to $k_{\sin}$

$$\frac{1}{a\cdot b}\frac{\partial k_{\sin}}{\partial c} = e^{-\frac{a^2+b^2}{2}}\,\frac{e^{abc} + e^{-abc}}{2} \tag{58}$$

which is equivalent to $k_{\cos}(a,b,c)$ (see Table 2 for detailed derivations).

For $\sigma(t) = \mathrm{erf}(t)$, the corresponding dual kernel is known as

$$k_{\mathrm{erf}}(a,b,c) = \frac{2}{\pi}\sin^{-1}\left(\frac{2abc}{\sqrt{(1+2a^2)(1+2b^2)}}\right). \tag{59}$$

Again, applying Theorem 3 to k_{erf} provides that

$$\begin{aligned}\frac{1}{a\cdot b}\frac{\partial k_{\mathrm{erf}}}{\partial c} &= \frac{2}{\pi}\frac{1}{a\cdot b}\frac{1}{\sqrt{1-\left(\frac{2abc}{\sqrt{(1+2a^2)(1+2b^2)}}\right)^2}}\cdot\frac{2ab}{\sqrt{(1+2a^2)(1+2b^2)}}\\ &= \frac{4}{\pi}\frac{1}{\sqrt{(1+2a^2)(1+2b^2) - 4a^2b^2c^2}}.\end{aligned} \tag{60}$$

One can check that this matches the dual kernel of $(\mathrm{erf}(t))' = \frac{2}{\sqrt{\pi}}e^{-t^2}$ from Table 2.

In addition, Theorem 3 holds for the ReLU activation because

$$\frac{1}{a \cdot b} \frac{\partial k_{\mathrm{ReLU}}}{\partial \theta} = \frac{1}{a \cdot b} \frac{\partial}{\partial c} \left(ab \frac{\sqrt{1-c^2} + (\pi - \cos^{-1}(c))c}{2\pi} \right) = \frac{\pi - \cos^{-1}(c)}{2\pi} \tag{61}$$

which is equivalent to the dual kernel of $\mathrm{ReLU}'(t) = \mathrm{Step}(t)$.

This theorem is used in `Elementwise` in our codebase to automatically derive the NTK given only the NNGP function.

C Proof of Theorem 4

Theorem 4 (Homogeneous NTK Embedding)**.** *Suppose that the dual kernel K_σ is homogeneous as per Equation (17). Also suppose $\widetilde{\kappa}(t)$ is a degree-q polynomial with non-negative coefficients that satisfies* $(\mathbf{1})$ $\max_{t\in[-1,1]} |\widetilde{\kappa}(t) - \kappa(t)| \leq \frac{1}{\mathrm{poly}(n)}$ *and* $\max_{t\in[-1,1]} |\widetilde{\kappa}'(t) - \kappa'(t)| \leq \frac{1}{\mathrm{poly}(n)}$, $(\mathbf{2})$ $\max_{|t|\leq 1+\frac{1}{\mathrm{poly}(n)}} |\widetilde{\kappa}(t+\gamma) - \widetilde{\kappa}(t)| \leq \frac{1}{\mathrm{poly}(n)}$ *and* $\max_{|t|\leq 1+\frac{1}{\mathrm{poly}(n)}} |\widetilde{\kappa}'(t+\gamma) - \widetilde{\kappa}'(t)| \leq \frac{1}{\mathrm{poly}(n)}$ *for any* $|\gamma| \leq \frac{1}{\mathrm{poly}(n)}$. *Then for any integer $L \geq 1$, any $\varepsilon, \lambda \geq \frac{1}{\mathrm{poly}(n)}$, and any dataset $\boldsymbol{X} \in \mathbb{R}^{d\times n}$ with $\|\boldsymbol{X}\|_F \leq \mathrm{poly}(n)$, if $\boldsymbol{K}_{\mathrm{ntk}} \in \mathbb{R}^{n\times n}$ is the depth-L NTK kernel matrix on this dataset, there exists $m = \mathcal{O}\left(\frac{s_\lambda(\boldsymbol{K}_{\mathrm{ntk}})}{\varepsilon^2} \cdot \mathrm{poly}\left(q^L, \log n\right)\right)$ such that the output $\psi^{(L)}(\boldsymbol{X}) \in \mathbb{R}^{m\times n}$ of Algorithm 1 satisfies with probability at least $1 - \frac{1}{\mathrm{poly}(n)}$*

$$(1-\varepsilon)\left(\boldsymbol{K}_{\mathrm{ntk}} + \lambda \boldsymbol{I}_n\right) \preceq \psi^{(L)}(\boldsymbol{X})^\top \psi^{(L)}(\boldsymbol{X}) + \lambda \boldsymbol{I}_n \preceq (1+\varepsilon)\left(\boldsymbol{K}_{\mathrm{ntk}} + \lambda \boldsymbol{I}_n\right). \tag{20}$$

Moreover, the runtime of Algorithm 1 is $\mathcal{O}\left(\mathrm{poly}\left(q^L, \log n\right) \cdot \varepsilon^{-2} \cdot (s_\lambda(\boldsymbol{K}_{\mathrm{ntk}}) \cdot n + \mathrm{nnz}(\boldsymbol{X}))\right)$.

Proof of Theorem 4: We start the proof by showing that the polynomial $R^{(L)}(t)$ defined as

$$R^{(L)}(t) := \sum_{h=0}^{L} \widetilde{\kappa}^{\circ h}(t) \cdot \prod_{i=h}^{L-1} \widetilde{\kappa}' \circ \widetilde{\kappa}^{\circ i}(t)$$

tightly approximates the following function at every point $t \in [-1, 1]$

$$T^{(L)}(t) := \sum_{h=0}^{L} \kappa^{\circ h}(t) \cdot \prod_{i=h}^{L-1} \kappa' \circ \kappa^{\circ i}(t)$$

Specifically, we prove that

$$\max_{t\in[-1,1]} \left| T^{(L)}(t) - R^{(L)}(t) \right| \leq \frac{1}{\mathrm{poly}\,(n)}. \tag{62}$$

In order to prove Equation (62), we first show that for every $h = 0, 1, 2, \ldots L$ the following holds

$$\max_{t\in[-1,1]} \left| \kappa^{\circ h}(t) - \widetilde{\kappa}^{\circ h}(t) \right| \leq \frac{1}{\mathrm{poly}\,(n)}.$$

The proof of the above is by induction on h. For $h = 0$ by convention $\kappa^{\circ h}(t) = \widetilde{\kappa}^{\circ h}(t) = t$, which proves the base of induction. For the inductive step suppose that $\max_{t\in[-1,1]} \left|\kappa^{\circ h-1}(t) - \widetilde{\kappa}^{\circ h-1}(t)\right| \leq \frac{1}{\mathrm{poly}(n)}$ holds for some $h \geq 1$. Using this inductive hypothesis along with preconditions of Theorem 4, for any $t \in [-1, 1]$ we can write,

$$\begin{aligned}
\left|\widetilde{\kappa}^{\circ h}(t) - \kappa^{\circ h}(t)\right| &\leq \left|\widetilde{\kappa}^{\circ h}(t) - \widetilde{\kappa} \circ \kappa^{\circ h-1}(t)\right| + \left|\widetilde{\kappa} \circ \kappa^{\circ h-1}(t) - \kappa^{\circ h}(t)\right| \\
&\leq \frac{1}{\mathrm{poly}\,(n)} + \left|\widetilde{\kappa} \circ \kappa^{\circ h-1}(t) - \kappa^{\circ h}(t)\right| \\
&\leq \frac{1}{\mathrm{poly}\,(n)},
\end{aligned}$$

where the first line above follows from triangle inequality. The second line above follows from precondition $(\mathbf{2})$ of the theorem. The third line follows from precondition $(\mathbf{1})$ of the theorem. Therefore $\max_{t\in[-1,1]} \left|\kappa^{\circ h}(t) - \widetilde{\kappa}^{\circ h}(t)\right| \leq \frac{1}{\mathrm{poly}(n)}$ for any $h = 0, 1, \ldots L$.

Moreover, by preconditions of the theorem, we can show in a similar fashion that $\max_{t\in[-1,1]} \left|\kappa' \circ \kappa^{\circ h-1}(t) - \widetilde{\kappa}' \circ \widetilde{\kappa}^{\circ h-1}(t)\right| \leq \frac{1}{\text{poly}(n)}$. These inequalities are sufficient to prove Equation (62).

Now, let us define the kernel $\widetilde{\Theta}_\sigma^{(L)}$ as

$$\widetilde{\Theta}_\sigma^{(L)}(x,y) := \|x\|_2 \|y\|_2 \cdot R^{(L)}\left(\frac{\langle x,y\rangle}{\|x\|_2 \|y\|_2}\right).$$

The depth-L NTK kernel, as we showed in Equation (19), is

$$\Theta_\sigma^{(L)}(x,y) := \|x\|_2 \|y\|_2 \cdot T^{(L)}\left(\frac{\langle x,y\rangle}{\|x\|_2 \|y\|_2}\right).$$

Using Equation (62), for any $x, y \in \mathbb{R}^d$, we have,

$$\left|\Theta_\sigma^{(L)}(x,y) - \widetilde{\Theta}_\sigma^{(L)}(x,y)\right| \leq \frac{\|x\|_2 \|y\|_2}{\text{poly}\,(n)}.$$

For any dataset $\boldsymbol{X} = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^{d\times n}$, we let $\widetilde{\boldsymbol{K}}_{\text{ntk}} \in \mathbb{R}^{n\times n}$ be the kernel matrix corresponding to the kernel function $\widetilde{\Theta}_\sigma^{(L)}$ and $\boldsymbol{X}$, i.e., $[\widetilde{\boldsymbol{K}}_{\text{ntk}}]_{i,j} = \widetilde{\Theta}_\sigma^{(L)}(x_i, x_j)$ to have that

$$\begin{aligned}\left\|\boldsymbol{K}_{\text{ntk}} - \widetilde{\boldsymbol{K}}_{\text{ntk}}\right\|_{op} &\leq \left\|\boldsymbol{K}_{\text{ntk}} - \widetilde{\boldsymbol{K}}_{\text{ntk}}\right\|_F \\ &\leq \frac{\|\boldsymbol{X}\|_F^2}{\text{poly}\,(n)} \\ &\leq \frac{1}{\text{poly}\,(n)} \leq \frac{\varepsilon\lambda}{3},\end{aligned}$$

where the third line above follows from the assumption of the theorem about $\|\boldsymbol{X}\|_F \leq \text{poly}\,(n)$ and $\varepsilon, \lambda \geq \frac{1}{\text{poly}(n)}$. Therefore, in order to prove the desired subspace embedding guarantee of Theorem 4, it suffices to prove that with probability at least $1 - \frac{1}{\text{poly}(n)}$, the following holds

$$(1-\varepsilon/2)\left(\widetilde{\boldsymbol{K}}_{\text{ntk}} + \lambda \boldsymbol{I}_n\right) \preceq \psi^{(L)}(\boldsymbol{X})^\top \psi^{(L)}(\boldsymbol{X}) + \lambda \boldsymbol{I}_n \preceq (1+\varepsilon/2)\left(\widetilde{\boldsymbol{K}}_{\text{ntk}} + \lambda \boldsymbol{I}_n\right).$$

From now on we focus on proving the above inequality. If we let $R^{(L)}(t) = \sum_{i=0}^p c_j t^j$ be the polynomial defined in line 3 of Algorithm 1 then we have that

$$\widetilde{\boldsymbol{K}}_{\text{ntk}} = \boldsymbol{D}\left(\sum_{j=0}^p c_j \left(\boldsymbol{Y}^{\otimes j}\right)^\top \boldsymbol{Y}^{\otimes j}\right)\boldsymbol{D} = \sum_{j=0}^p c_j \cdot \left(\boldsymbol{Y}^{\otimes j}\boldsymbol{D}\right)^\top \boldsymbol{Y}^{\otimes j}\boldsymbol{D}$$

where

$$\boldsymbol{D} = \text{diag}\left([\|x_1\|_2, \ldots, \|x_n\|_2]\right) \in \mathbb{R}^{n\times n}, \quad \boldsymbol{Y} = \left[\frac{x_1}{\|x_1\|_2}, \ \ldots, \frac{x_n}{\|x_n\|_2}\right] \in \mathbb{R}^{d\times n}.$$

Note that each of the term $\left(\boldsymbol{Y}^{\otimes j}\boldsymbol{D}\right)^\top \boldsymbol{Y}^{\otimes j}\boldsymbol{D} = \boldsymbol{D}(\boldsymbol{Y}^{\otimes j})^\top \boldsymbol{Y}^{\otimes j}\boldsymbol{D}$ is a positive definite Gram matrix. Also, from the fact that coefficients c_j are positive and by Courant-Fischer's min-max theorem, the statistical dimension of the Gram matrix $c_j \cdot \left(\boldsymbol{Y}^{\otimes j}\boldsymbol{D}\right)^\top \boldsymbol{Y}^{\otimes j}\boldsymbol{D}$ for every $j \geq 0$ is upper bounded by the statistical dimension of the kernel matrix $\widetilde{\boldsymbol{K}}_{\text{ntk}}$. More specifically, for any $\mu > 0$ and every $j = 0, 1, \ldots p$, we have

$$s_\mu\left(c_j \cdot \left(\boldsymbol{Y}^{\otimes j}\boldsymbol{D}\right)^\top \boldsymbol{Y}^{\otimes j}\boldsymbol{D}\right) \leq s_\mu\left(\widetilde{\boldsymbol{K}}_{\text{ntk}}\right).$$

Now let $\mu := \frac{\lambda}{p+1}$ and note that from the definition of statistical dimension it follows that $s_\mu\left(\widetilde{\boldsymbol{K}}_{\text{ntk}}\right) \leq (p+1)s_\lambda\left(\widetilde{\boldsymbol{K}}_{\text{ntk}}\right)$. The sketch matrix Q^j defined in line 4 of the algorithm has

$m = \Omega\left(\varepsilon^{-2} s_\lambda(\boldsymbol{K}_{\mathrm{ntk}}) \cdot \mathrm{poly}\left(q^L, \log n\right)\right) = \Omega\left(\varepsilon^{-2} s_\mu(\boldsymbol{K}_{\mathrm{ntk}}) \cdot \mathrm{poly}\left(q^L, \log n\right)\right)$ rows. Therefore, by Theorem 5, the following holds for every $j = 0, 1, \ldots p$ with probability at least $1 - \frac{1}{\mathrm{poly}(n)}$,

$$\frac{c_j \cdot \boldsymbol{D}(\boldsymbol{Y}^{\otimes j})^\top \boldsymbol{Y}^{\otimes j} \boldsymbol{D} + \mu \boldsymbol{I}_n}{1+\varepsilon/3} \preceq c_j \cdot \boldsymbol{D}\left(Q^j \boldsymbol{Y}^{\otimes j}\right)^\top Q^j \boldsymbol{Y}^{\otimes j} \boldsymbol{D} + \mu \boldsymbol{I}_n \preceq \frac{c_j \cdot \boldsymbol{D}(\boldsymbol{Y}^{\otimes j})^\top \boldsymbol{Y}^{\otimes j} \boldsymbol{D} + \mu \boldsymbol{I}_n}{1-\varepsilon/3}.$$

By union bound over $p+1 = \mathcal{O}(q^L) = o(\mathrm{poly}\,(n))$ events, the above inequality holds simultaneously for all j with high probability in n. Thus, by summing up the above inequality over all j and using the fact that $\mu = \frac{\lambda}{p+1}$ we find that,

$$\frac{\widetilde{\boldsymbol{K}}_{\mathrm{ntk}} + \lambda \boldsymbol{I}_n}{1+\varepsilon/3} \preceq \sum_{j=0}^{p} \left(c_j \cdot \boldsymbol{D}^j \left(Q^j \boldsymbol{Y}^{\otimes j}\right)^\top Q^j \boldsymbol{Y}^{\otimes j} \boldsymbol{D}\right) + \lambda \boldsymbol{I}_n \preceq \frac{\widetilde{\boldsymbol{K}}_{\mathrm{ntk}} + \lambda \boldsymbol{I}_n}{1-\varepsilon/3}.$$

This proves the theorem because the output of the algorithm satisfies $\psi^{(L)}(\boldsymbol{X})^\top \psi^{(L)}(\boldsymbol{X}) = \sum_{j=0}^{p} c_j \boldsymbol{D}^j \left(Q^j \boldsymbol{Y}^{\otimes j}\right)^\top Q^j \boldsymbol{Y}^{\otimes j} \boldsymbol{D}^j$. The runtime bound follows immediately from Theorem 5. □

D Convolutional Neural Tangent Kernel

In this section, we design and analyze an efficient oblivious sketch for the Convolutional Neural Tangent Kernel (CNTK), which is the kernel function corresponding to a CNN with infinite number of channels. Arora et al. [9] gave dynamic programming (DP) based solutions for computing two variants of CNTK; one is the vanilla version which performs no pooling, and the other performs Global Average Pooling (GAP) on its top layer. For conciseness, we focus mainly on the CNTK with GAP, which also exhibits superior empirical performance [9]. However, we remark that the vanilla CNTK has a very similar structure and hence our techniques can be applied to it, as well.

We start by restating the DP approach proposed in [9] for computing the L-layered CNTK with an arbitrary activation function σ, convolutional filters of size $q \times q$ and GAP. Consider two input images $y, z \in \mathbb{R}^{d_1 \times d_2 \times c}$ where c is the number of channels ($c = 3$ for the standard color image).

1. For every $i, i' \in [d_1]$ and $j, j' \in [d_2]$, define

$$\Gamma^{(0)}_{i,j,i',j'}(y,z) := \sum_{l=1}^{c} y_{i,j,l} \cdot z_{i',j',l}, \tag{63}$$
$$K^{(0)}_{i,j,i',j'}(y,z) := \sum_{a=-\frac{q-1}{2}}^{\frac{q-1}{2}} \sum_{b=-\frac{q-1}{2}}^{\frac{q-1}{2}} \Gamma^{(0)}_{i+a,j+b,i'+a,j'+b}(y,z).$$

2. For every $h \in [L]$, every $i, i' \in [d_1]$ and $j, j' \in [d_2]$, define

$$\Gamma^{(h)}_{i,j,i',j'}(y,z) := \frac{1}{q^2} \cdot \mathop{\mathbb{E}}_{(u,v)\sim\mathcal{N}\left(0, \boldsymbol{\Lambda}^{(h)}_{i,j,i',j'}(x,y)\right)} \left[\sigma(u)\sigma(v)\right],$$
$$K^{(h)}_{i,j,i',j'}(y,z) := \sum_{a=-\frac{q-1}{2}}^{\frac{q-1}{2}} \sum_{b=-\frac{q-1}{2}}^{\frac{q-1}{2}} \Gamma^{(h)}_{i+a,j+b,i'+a,j'+b}(y,z), \tag{64}$$

   where the covariance matrix is

$$\boldsymbol{\Lambda}^{(h)}_{i,j,i',j'}(x,y) := \begin{bmatrix} K^{(h-1)}_{i,j,i',j'}(y,y) & K^{(h-1)}_{i,j,i',j'}(y,z) \\ K^{(h-1)}_{i,j,i',j'}(z,y) & K^{(h-1)}_{i,j,i',j'}(z,z) \end{bmatrix} \in \mathbb{R}^{2\times 2}. \tag{65}$$

3. For every $h \in [L]$, every $i, i' \in [d_1]$ and $j, j' \in [d_2]$, define

$$\dot{\Gamma}^{(h)}_{i,j,i',j'}(y,z) := \frac{1}{q^2} \cdot \mathop{\mathbb{E}}_{(u,v)\sim\mathcal{N}\left(0, \boldsymbol{\Lambda}^{(h)}_{i,j,i',j'}(y,z)\right)} \left[\sigma'(u)\sigma'(v)\right]. \tag{66}$$

4. Let $\Pi^{(0)}(x,y) := 0$ and for every $h \in [L-1]$, every $i, i' \in [d_1]$ and $j, j' \in [d_2]$, define

$$\Pi^{(h)}_{i,j,i',j'}(y,z) := \sum_{a=-\frac{q-1}{2}}^{\frac{q-1}{2}} \sum_{b=-\frac{q-1}{2}}^{\frac{q-1}{2}} \left[\Pi^{(h-1)}(y,z) \odot \dot{\Gamma}^{(h)}(y,z) + \Gamma^{(h)}(y,z)\right]_{i+a,j+b,i'+a,j'+b}, \tag{67}$$

and also $\Pi^{(L)}(y,z) := \Pi^{(L-1)}(y,z) \odot \dot{\Gamma}^{(L)}(y,z)$.

5. The final CNTK expressions is defined as:

$$\Theta^{(L)}_{\text{cntk}}(y,z) := \frac{1}{d_1^2 d_2^2} \cdot \sum_{i,i' \in [d_1]} \sum_{j,j' \in [d_2]} \Pi^{(L)}_{i,j,i',j'}(y,z). \tag{68}$$

The above procedure for exact computation of the depth-L CNTK value $\Theta^{(L)}_{\text{cntk}}(y,z)$ takes $\Omega\left((d_1 d_2)^2 (c+L)\right)$ runtime, which is extremely slow particularly due to its quadratic dependence on the number of pixels of input images $d_1 d_2$. Fortunately, we are able to show that the CNTK for homogeneous dual kernels, as per Equation (17), is a highly structured object that can be fully characterized in terms of tensoring and composition of the dot-product factor of dual kernels, and exploiting this special structure is key in designing efficient sketching methods for the CNTK.

D.1 CNTK for Homogeneous Dual Kernels

In this section we show that the CNTK function corresponding to any homogeneous dual kernel, i.e., $K_\sigma(x,y) = \|x\|_2 \|y\|_2 \cdot \kappa\left(\frac{\langle x,y\rangle}{\|x\|_2 \|y\|_2}\right)$ for some $\kappa : [-1,1] \to [-1,1]$, takes a simple form which enables us to devise efficient sketching algorithms for the CNTK. Unlike the fully-connected NTK, the CNTK is not a simple dot-product kernel function. The key reason being that CNTK works by partitioning its input images into patches and locally transforming the patches at each layer, as opposed to the NTK which operates on the entire input vectors. The depth-L CNTK corresponding to homogeneous dual kernels can be fully characterized in terms of tensoring and composition of the dot-product kernel κ and its derivative κ'.

Definition 2 (CNTK for Homogeneous Dual Kernels)**.** For every positive integers q, L, the L-layered CNTK for a homogeneous dual kernel, as per Equation (17), and convolutional filter size of $q \times q$ is defined as follows

1. For $x \in \mathbb{R}^{d_1 \times d_2 \times c}$, every $i \in [d_1]$ and $j \in [d_2]$ let $N^{(0)}_{i,j}(x) := q^2 \cdot \sum_{l=1}^{c} |x_{i,j,l}|^2$, and for every $h \geq 1$, recursively define,

$$N^{(h)}_{i,j}(x) := \frac{1}{q^2} \cdot \sum_{a=-\frac{q-1}{2}}^{\frac{q-1}{2}} \sum_{b=-\frac{q-1}{2}}^{\frac{q-1}{2}} N^{(h-1)}_{i+a,j+b}(x). \tag{69}$$

2. For every $h \in [h]$, every $i, i' \in [d_1]$ and $j, j' \in [d_2]$, define

$$\Gamma^{(h)}_{i,j,i',j'}(y,z) := \frac{\sqrt{N^{(h)}_{i,j}(y) \cdot N^{(h)}_{i',j'}(z)}}{q^2} \cdot \kappa\left(A\right), \quad \Gamma^{(0)}_{i,j,i',j'}(y,z) = \sum_{l=1}^{c} y_{i,j,l} \cdot z_{i',j',l} \tag{70}$$

where $A = \frac{1}{\sqrt{N^{(h)}_{i,j}(y) \cdot N^{(h)}_{i',j'}(z)}} \sum_{a=-\frac{q-1}{2}}^{\frac{q-1}{2}} \sum_{b=-\frac{q-1}{2}}^{\frac{q-1}{2}} \Gamma^{(h-1)}_{i+a,j+b,i'+a,j'+b}(y,z)$.

3. For every $h \in [L]$, every $i, i' \in [d_1]$ and $j, j' \in [d_2]$, define

$$\dot{\Gamma}^{(h)}_{i,j,i',j'}(y,z) := \frac{1}{q^2} \cdot \kappa'\left(A\right). \tag{71}$$

4. Let $\Pi^{(0)}(y,z) := 0$ and for every $h \in [L-1]$, every $i, i' \in [d_1]$ and $j, j' \in [d_2]$, define

$$\Pi^{(h)}_{i,j,i',j'}(y,z) := \sum_{a=-\frac{q-1}{2}}^{\frac{q-1}{2}} \sum_{b=-\frac{q-1}{2}}^{\frac{q-1}{2}} \left[\Pi^{(h-1)}(y,z) \odot \dot{\Gamma}^{(h)}(y,z) + \Gamma^{(h)}(y,z)\right]_{i+a,j+b,i'+a,j'+b}, \tag{72}$$

Furthermore, define

$$\Pi^{(L)}(y,z) := \Pi^{(L-1)}(y,z) \odot \dot{\Gamma}^{(L)}(y,z). \tag{73}$$

5. The final CNTK expressions for ReLU activation is:

$$\Theta^{(L)}_{\text{cntk}}(y,z) := \frac{1}{d_1^2 d_2^2} \cdot \sum_{i,i' \in [d_1]} \sum_{j,j' \in [d_2]} \Pi^{(L)}_{i,j,i',j'}(y,z). \tag{74}$$

We now describes some of the basic properties of the functions $\Gamma^{(h)}(y,z), \dot{\Gamma}^{(h)}(y,z)$, and $\Pi^{(h)}(y,z)$ defined in Equation (70), in the following lemma,

Lemma 4 (Properties of $\Gamma^{(h)}(y,z), \dot{\Gamma}^{(h)}(y,z)$, and $\Pi^{(h)}(y,z)$)**.** *Suppose that the dot-product kernel $\kappa(\cdot)$ in Equation (17) and its derivative satisfy $\kappa(1) = \kappa'(1) = 1$. For every images $y, z \in \mathbb{R}^{d_1 \times d_2 \times c}$, every integer $h \geq 0$ and every $i, i' \in [d_1]$ and $j, j' \in [d_2]$ the following properties are satisfied by functions $\Gamma^{(h)}, \dot{\Gamma}^{(h)}, \Pi^{(h)}$ and $N^{(h)}$ defined in Equation (70), Equation (71), Equation (72) and Equation (73), and Equation (69) of Definition 2:*

1. **Cauchy–Schwarz:** $\left|\Gamma^{(h)}_{i,j,i',j'}(y,z)\right| \leq \frac{\sqrt{N^{(h)}_{i,j}(y) \cdot N^{(h)}_{i',j'}(z)}}{q^2}$, *and* $\left|\dot{\Gamma}^{(h)}_{i,j,i',j'}(y,z)\right| \leq \frac{1}{q^2}$, *and* $\Pi^{(h)}_{i,j,i',j'}(y,z) \leq \sqrt{\Pi^{(h)}_{i,j,i,j}(y,y) \cdot \Pi^{(h)}_{i',j',i',j'}(z,z)}$.

2. **Norm value:** $\Gamma^{(h)}_{i,j,i,j}(y,y) = \frac{N^{(h)}_{i,j}(y)}{q^2} \geq 0$, *and* $\dot{\Gamma}^{(h)}_{i,j,i,j}(y,y) = \frac{1}{q^2} \geq 0$, *and* $\Pi^{(h)}_{i,j,i,j}(y,y) = \begin{cases} h \cdot N^{(h+1)}_{i,j}(y) & \text{if } h < L \\ \frac{L-1}{q^2} \cdot N^{(L)}_{i,j}(y) & \text{if } h = L \end{cases}$.

The properties stated in the above lemma can be straightforwardly proved using induction.

D.2 CNTK Sketch for Homogeneous Dual Kernels

Our sketching method relies on approximating the dot-product kernel function $\kappa(\cdot)$ and its derivative $\kappa'(\cdot)$ with low-degree polynomials via Taylor expansion, and then applying PolySketch to the resulting polynomial kernels. Our sketch computes the features for each pixel of the input image, by tensor product of the sketches for function $\kappa(\cdot)$ at consecutive layers, which in turn can be sketched efficiently by PolySketch. Additionally, the features of pixels that lie in the same patch get *locally combined* at each layer via direct sum operation. This precisely corresponds to the convolution operation in neural networks. We start by presenting our CNTK Sketch algorithm in Algorithm 4 and prove the correctness and runtime of our procedure in Theorem 6.

Theorem 6 (Correctness and Runtime of Algorithm 4)**.** *Suppose that the dual kernel K_σ is homogeneous as per Equation (17) also assume that $\kappa(1) = \kappa'(1) = 1$. Fix some $\varepsilon > 0$ and $L \in \mathbb{Z}_{>0}$ and suppose that $\widetilde{\kappa}(t)$ and $\widetilde{\kappa}'(t)$ are degree-p polynomials with non-negative coefficients that satisfies* (**1**) $\max_{t \in [-1,1]} |\widetilde{\kappa}(t) - \kappa(t)| = \mathcal{O}\left(\frac{\varepsilon}{L}\right)$ *and* $\max_{t \in [-1,1]} |\widetilde{\kappa}'(t) - \kappa'(t)| = \mathcal{O}\left(\frac{\varepsilon}{L}\right)$, (**2**) $\max_{|t| \leq 1+\mathcal{O}(\varepsilon)} |\widetilde{\kappa}(t+\gamma) - \widetilde{\kappa}(t)| \leq \mathcal{O}(\gamma)$ *and* $\max_{|t| \leq 1+\mathcal{O}(\varepsilon)} |\widetilde{\kappa}'(t+\gamma) - \widetilde{\kappa}'(t)| \leq \mathcal{O}(\gamma)$ *for any* $|\gamma| \leq \mathcal{O}(\varepsilon)$. *If $m = \Omega\left(\frac{L^4 p}{\varepsilon^2} \cdot \log^3 n\right)$ and $m' = \Omega\left(\frac{L^2}{\varepsilon^2} \cdot \log^3 n\right)$, then for any $y, z \in \mathbb{R}^{d_1 \times d_2 \times c}$, the output of Algorithm 4 satisfies*

$$\Pr\left[\left|\left\langle \Psi^{(L)}_{\text{cntk}}(y), \Psi^{(L)}_{\text{cntk}}(z) \right\rangle - \Theta^{(L)}_{\text{cntk}}(y,z)\right| > \varepsilon \cdot \sqrt{\Theta^{(L)}_{\text{cntk}}(y,y) \cdot \Theta^{(L)}_{\text{cntk}}(z,z)}\right] \leq \frac{1}{\text{poly}(n)}.$$

Furthermore, for every image $x \in \mathbb{R}^{d_1 \times d_2 \times c}$, $\Psi^{(L)}_{\text{cntk}}(x) \in \mathbb{R}^{m'}$ can be computed in time $\mathcal{O}\left(Lp^2 m \log m \cdot d_1 d_2\right)$.

Proof. The correctness proof is by induction on the value of $h = 0, 1, 2, \ldots L$. More formally, consider the following invariants for every iteration $h = 0, 1, 2, \ldots L$ of the algorithm:

Algorithm 4 CNTK Sketch for Homogeneous Dual Kernels

input: image $x \in \mathbb{R}^{d_1 \times d_2 \times c}$, depth L, filter size q, sketching dimensions m, m', polynomials $\widetilde{\kappa}(t) = \sum_{j=0}^{p} a_j t^j$ and $\widetilde{\kappa}'(t) = \sum_{j=0}^{p} b_j t^j$ with $a_j, b_j \in \mathbb{R}_+$

for every $i \in [d_1]$, $j \in [d_2]$, and $h = 0, 1, 2, \ldots L$ compute $N_{i,j}^{(h)}(x)$ as per Equation (69)

for every $i \in [d_1]$, $j \in [d_2]$, initialize $\phi_{i,j}^{(0)}(x) \leftarrow x_{i,j,:}$ and $\psi_{i,j}^{(0)}(x) \leftarrow 0$

for $h = 1$ to L **do**

For $\ell = 0, \ldots, p$, let Q^ℓ be a degree-ℓ PolySketch with target dimension m and for every $i \in [d_1]$, $j \in [d_2]$ compute

$$Z_{i,j,\ell}^{(h)}(x) \leftarrow Q^\ell \cdot \left(\mu_{i,j}^{(h)}(x)\right)^{\otimes \ell}, \quad \mu_{i,j}^{(h)}(x) \leftarrow \frac{1}{\sqrt{N_{i,j}^{(h)}(x)}} \cdot \bigoplus_{a=-\frac{q-1}{2}}^{\frac{q-1}{2}} \bigoplus_{b=-\frac{q-1}{2}}^{\frac{q-1}{2}} \phi_{i+a,j+b}^{(h-1)}(x)$$

for every $i \in [d_1]$, $j \in [d_2]$ construct $\phi_{i,j}^{(h)}(x) \leftarrow \frac{\sqrt{N_{i,j}^{(h)}(x)}}{q} \cdot \bigoplus_{\ell=0}^{p} \sqrt{a_\ell} \cdot Z_{i,j,\ell}^{(h)}(x)$

for every $i \in [d_1]$, $j \in [d_2]$ construct $\dot{\phi}^{(h)}(x) \leftarrow \frac{1}{q} \cdot \bigoplus_{\ell=0}^{p} \sqrt{b_\ell} \cdot Z_{i,j,\ell}^{(h)}(x)$

Let Q^2 be a degree-2 PolySketch with target dimension m'

if $h = L$ **then**

for every $i \in [d_1]$, $j \in [d_2]$ compute

$$\psi_{i,j}^{(h)}(x) \leftarrow \bigoplus_{a=-\frac{q-1}{2}}^{\frac{q-1}{2}} \bigoplus_{b=-\frac{q-1}{2}}^{\frac{q-1}{2}} \left[Q^2\left(\psi_{i+a,j+b}^{(h-1)}(x) \otimes \dot{\phi}_{i+a,j+b}^{(h)}(x)\right) \oplus \phi_{i+a,j+b}^{(h)}(x)\right]$$

else

for every $i \in [d_1]$ and $j \in [d_2]$ compute

$$\psi_{i,j}^{(L)}(x) \leftarrow \bigoplus_{a=-\frac{q-1}{2}}^{\frac{q-1}{2}} \bigoplus_{b=-\frac{q-1}{2}}^{\frac{q-1}{2}} Q^2\left(\psi_{i+a,j+b}^{(L-1)}(x) \otimes \dot{\phi}_{i+a,j+b}^{(L)}(x)\right)$$

return $\Psi_{\mathrm{cntk}}^{(L)}(y,z) := \frac{1}{d_1 d_2} \cdot \sum_{i \in [d_1]} \sum_{j \in [d_2]} \psi_{i,j}^{(L)}(x)$

$\mathbf{P_1}(\mathbf{h})$: Simultaneously for all $i, i' \in [d_1]$ and $j, j' \in [d_2]$:

$$\left|\left\langle \phi_{i,j}^{(h)}(y), \phi_{i',j'}^{(h)}(z)\right\rangle - \Gamma_{i,j,i',j'}^{(h)}(y,z)\right| \le (h+1) \cdot \frac{\varepsilon}{60L^2} \cdot \frac{\sqrt{N_{i,j}^{(h)}(y) \cdot N_{i',j'}^{(h)}(z)}}{q^2},$$

$$\left|\left\|\phi_{i,j}^{(h)}(y)\right\|_2^2 - \Gamma_{i,j,i,j}^{(h)}(y,y)\right| \le \frac{(h+1)\cdot\varepsilon}{60L^2} \cdot \frac{N_{i,j}^{(h)}(y)}{q^2},$$

$$\left|\left\|\phi_{i',j'}^{(h)}(z)\right\|_2^2 - \Gamma_{i',j',i',j'}^{(h)}(z,z)\right| \le \frac{(h+1)\cdot\varepsilon}{60L^2} \cdot \frac{N_{i',j'}^{(h)}(z)}{q^2}.$$

$\mathbf{P_2}(\mathbf{h})$: Simultaneously for all $i, i' \in [d_1]$ and $j, j' \in [d_2]$:

$$\left|\left\langle \psi_{i,j}^{(h)}(y), \psi_{i',j'}^{(h)}(z)\right\rangle - \Pi_{i,j,i',j'}^{(h)}(y,z)\right| \le \begin{cases} \frac{\varepsilon}{10} \cdot \frac{h^2}{L+1} \cdot \sqrt{N_{i,j}^{(h+1)}(y) \cdot N_{i',j'}^{(h+1)}(z)} & \text{if } h < L \\ \frac{\varepsilon}{10} \cdot \frac{L-1}{q^2} \cdot \sqrt{N_{i,j}^{(L)}(y) \cdot N_{i',j'}^{(L)}(z)} & \text{if } h = L \end{cases},$$

$$(\text{only for } h < L): \quad \left|\left\|\psi_{i,j}^{(h)}(y)\right\|_2^2 - \Pi_{i,j,i,j}^{(h)}(y,y)\right| \le \frac{\varepsilon}{10} \cdot \frac{h^2}{L+1} \cdot N_{i,j}^{(h+1)}(y),$$

$$(\text{only for } h < L): \quad \left|\left\|\psi_{i',j'}^{(h)}(z)\right\|_2^2 - \Pi_{i',j',i',j'}^{(h)}(z,z)\right| \le \frac{\varepsilon}{10} \cdot \frac{h^2}{L+1} \cdot N_{i',j'}^{(h+1)}(z).$$

We prove that probabilities $\Pr[P_1(0)]$ and $\Pr[P_2(0)|P_1(0)]$ are both greater than $1-\frac{1}{\text{poly}(n)}$. Additionally, for every $h=1,2,\dots L$, we prove that the conditional probabilities $\Pr[P_1(h)|P_1(h-1)]$ and $\Pr[P_2(h)|P_2(h-1),P_1(h),P_1(h-1)]$ are greater than $1-\frac{1}{\text{poly}(n)}$. These invariants immediately give the correctness proof.

The **base of induction** corresponds to $h=0$. By line 3 of the algorithm, $\phi^{(0)}_{i,j}(y)=y_{i,j,:}$ and $\phi^{(0)}_{i',j'}(z)=z_{i',j',:}$, therefore, by using Equation (70), it trivially holds that $\Pr[P_1(0)]=1\geq 1-\frac{1}{\text{poly}(n)}$. Moreover, by line 3, we have that $\psi^{(0)}_{i,j}(y)=0$ and $\psi^{(0)}_{i',j'}(z)=0$, thus, by Equation (72), it trivially holds that $\Pr[P_2(0)|P_1(0)]=1\geq 1-\frac{1}{\text{poly}(n)}$. This completes the base of induction.

We now proceed to prove the **inductive step**. By assuming the inductive hypothesis for $h-1$, we prove that statements $P_1(h)$ and $P_2(h)$ hold. More precisely, first we condition on the statement $P_1(h-1)$ being true for some $h\geq 1$, and then prove that $P_1(h)$ holds with probability at least $1-\frac{1}{\text{poly}(n)}$. Next we show that conditioned on statements $P_2(h-1),P_1(h),P_1(h-1)$ being true, $P_2(h)$ holds with probability at least $1-\frac{1}{\text{poly}(n)}$. This will complete the induction.

First, by conditioning on the inductive hypothesis $P_1(h-1)$ and using the definition of $\mu^{(h)}_{i,j}(\cdot)$ in line 5 of the algorithm and applying Cauchy–Schwarz inequality and invoking Lemma 4 we find that,

$$\begin{aligned}
&\left|\left\langle \mu^{(h)}_{i,j}(y),\mu^{(h)}_{i',j'}(z)\right\rangle - \frac{\sum_{a=-\frac{q-1}{2}}^{\frac{q-1}{2}}\sum_{b=-\frac{q-1}{2}}^{\frac{q-1}{2}}\Gamma^{(h-1)}_{i+a,j+b,i'+a,j'+b}(y,z)}{\sqrt{N^{(h)}_{i,j}(y)\cdot N^{(h)}_{i',j'}(z)}}\right| \\
&\leq \frac{\sum_{a=-\frac{q-1}{2}}^{\frac{q-1}{2}}\sum_{b=-\frac{q-1}{2}}^{\frac{q-1}{2}}\sqrt{N^{(h-1)}_{i+a,j+b}(y)\cdot N^{(h-1)}_{i'+a,j'+b}(z)}}{q^2\cdot\sqrt{N^{(h)}_{i,j}(y)\cdot N^{(h)}_{i',j'}(z)}}\cdot\frac{h\cdot\varepsilon}{60L^2} \\
&\leq \frac{\sqrt{\sum_{a=-\frac{q-1}{2}}^{\frac{q-1}{2}}\sum_{b=-\frac{q-1}{2}}^{\frac{q-1}{2}}N^{(h-1)}_{i+a,j+b}(y)/q^2}\cdot\sqrt{\sum_{a=-\frac{q-1}{2}}^{\frac{q-1}{2}}\sum_{b=-\frac{q-1}{2}}^{\frac{q-1}{2}}N^{(h-1)}_{i'+a,j'+b}(z)/q^2}}{\sqrt{N^{(h)}_{i,j}(y)\cdot N^{(h)}_{i',j'}(z)}}\cdot\frac{h\cdot\varepsilon}{60L^2} \\
&= h\cdot\frac{\varepsilon}{60L^2},
\end{aligned} \tag{75}$$

where the last line follows from Equation (69).

Furthermore, if we let the collection of vectors $\left\{Z^{(h)}_{i,j,\ell}(y)\right\}_{\ell=0}^{p}$ and $\left\{Z^{(h)}_{i,j,\ell}(z)\right\}_{\ell=0}^{p}$ be defined as per line 5 of the algorithm, then by Theorem 5 and union bound, the following inequalities hold, with probability at least $1-\frac{1}{\text{poly}(n)}$, simultaneously for all $\ell=0,1,2,\dots p$, all $i,i'\in[d_1]$ and $j,j'\in[d_2]$:

$$\begin{aligned}
&\left|\left\langle Z^{(h)}_{i,j,\ell}(y),Z^{(h)}_{i',j',\ell}(z)\right\rangle - \left\langle \mu^{(h)}_{i,j}(y),\mu^{(h)}_{i',j'}(z)\right\rangle^{\ell}\right| \leq \mathcal{O}\left(\frac{\varepsilon}{L^2}\right)\left\|\mu^{(h)}_{i,j}(y)\right\|_2^{\ell}\left\|\mu^{(h)}_{i',j'}(z)\right\|_2^{\ell} \\
&\left\|Z^{(h)}_{i,j,\ell}(y)\right\|_2^2 \leq \frac{11}{10}\cdot\left\|\mu^{(h)}_{i,j}(y)\right\|_2^{2\ell} \\
&\left\|Z^{(h)}_{i',j',\ell}(z)\right\|_2^2 \leq \frac{11}{10}\cdot\left\|\mu^{(h)}_{i',j'}(z)\right\|_2^{2\ell}
\end{aligned} \tag{76}$$

Therefore, by Cauchy–Schwarz inequality, we find that with probability at least $1-\frac{1}{\text{poly}(n)}$, the following holds simultaneously for all $i,i'\in[d_1]$ and $j,j'\in[d_2]$:

$$\left|\left\langle \phi^{(h)}_{i,j}(y),\phi^{(h)}_{i',j'}(z)\right\rangle - \frac{\sqrt{N^{(h)}_{i,j}(y)N^{(h)}_{i',j'}(z)}}{q^2}\cdot\widetilde{\kappa}\left(\left\langle \mu^{(h)}_{i,j}(y),\mu^{(h)}_{i',j'}(z)\right\rangle\right)\right| \leq \mathcal{O}\left(\frac{\varepsilon}{L^2}\right)\cdot B, \tag{77}$$

where $B := \frac{\sqrt{N^{(h)}_{i,j}(y)N^{(h)}_{i',j'}(z)}}{q^2}\cdot\sqrt{\widetilde{\kappa}\left(\|\mu^{(h)}_{i,j}(y)\|_2^2\right)\cdot\widetilde{\kappa}\left(\|\mu^{(h)}_{i',j'}(z)\|_2^2\right)}$.

By conditioning on the inductive hypothesis $P_1(h-1)$ and using Lemma 4 we have,

$$\left|\left\|\mu_{i,j}^{(h)}(y)\right\|_2^2 - 1\right| \leq h \cdot \frac{\varepsilon}{60L^2}, \text{ and } \left|\left\|\mu_{i',j'}^{(h)}(z)\right\|_2^2 - 1\right| \leq h \cdot \frac{\varepsilon}{60L^2}.$$

Therefore, the precondition of the theorem implies that $\left|\widetilde{\kappa}\left(\|\mu_{i,j}^{(h)}(y)\|_2^2\right) - \widetilde{\kappa}(1)\right| \leq h \cdot \frac{\varepsilon}{60L^2}$ and $\left|\widetilde{\kappa}\left(\|\mu_{i',j'}^{(h)}(z)\|_2^2\right) - \widetilde{\kappa}(1)\right| \leq h \cdot \frac{\varepsilon}{60L^2}$. Consequently, because $\widetilde{\kappa}(1) \leq 1.01\kappa(1) = 1.01$, we find that

$$B \leq \frac{11}{10} \cdot \frac{\sqrt{N_{i,j}^{(h)}(y) N_{i',j'}^{(h)}(z)}}{q^2}.$$

By plugging this into Equation (77) we find that the following holds simultaneously for all $i, i' \in [d_1]$ and all $j, j' \in [d_2]$, with probability at least $1 - \frac{1}{\text{poly}(n)}$,

$$\left|\left\langle \phi_{i,j}^{(h)}(y), \phi_{i',j'}^{(h)}(z)\right\rangle - \frac{\sqrt{N_{i,j}^{(h)}(y) N_{i',j'}^{(h)}(z)}}{q^2} \cdot \widetilde{\kappa}\left(\left\langle \mu_{i,j}^{(h)}(y), \mu_{i',j'}^{(h)}(z)\right\rangle\right)\right| \leq \mathcal{O}\left(\frac{\varepsilon}{L^2}\right) \cdot \frac{\sqrt{N_{i,j}^{(h)}(y) N_{i',j'}^{(h)}(z)}}{q^2}. \tag{78}$$

We recall that $A := \frac{1}{\sqrt{N_{i,j}^{(h)}(y) \cdot N_{i',j'}^{(h)}(z)}} \sum_{a=-\frac{q-1}{2}}^{\frac{q-1}{2}} \sum_{b=-\frac{q-1}{2}}^{\frac{q-1}{2}} \Gamma_{i+a,j+b,i'+a,j'+b}^{(h-1)}(y,z)$ and

$$\Gamma_{i,j,i',j'}^{(h)}(y,z) = \frac{\sqrt{N_{i,j}^{(h)}(y) N_{i',j'}^{(h)}(z)}}{q^2} \kappa(A).$$

Note that by Lemma 4 and Equation (69), $-1 \leq A \leq 1$. Hence, using the precondition of the theorem and Equation (75) to find that,

$$\left|\widetilde{\kappa}\left(\left\langle \mu_{i,j}^{(h)}(y), \mu_{i',j'}^{(h)}(z)\right\rangle\right) - \widetilde{\kappa}(A)\right| \leq h \cdot \frac{\varepsilon}{60L^2}.$$

By incorporating the above inequality into Equation (78) using triangle inequality we find that, with probability at least $1 - \frac{1}{\text{poly}(n)}$, the following holds simultaneously for all $i, i' \in [d_1]$ and all $j, j' \in [d_2]$:

$$\left|\left\langle \phi_{i,j}^{(h)}(y), \phi_{i',j'}^{(h)}(z)\right\rangle - \frac{\sqrt{N_{i,j}^{(h)}(y) N_{i',j'}^{(h)}(z)}}{q^2} \cdot \widetilde{\kappa}(A)\right| \leq \left(\mathcal{O}\left(\frac{\varepsilon}{L^2}\right) + \frac{h \cdot \varepsilon}{60L^2}\right) \cdot \frac{\sqrt{N_{i,j}^{(h)}(y) N_{i',j'}^{(h)}(z)}}{q^2}. \tag{79}$$

Additionally, since $-1 \leq A \leq 1$, using the preconditions of the theorem we can conclude that $|\widetilde{\kappa}(A) - \kappa(A)| \leq \frac{\varepsilon}{76L^2}$. By combining the above inequality with Equation (79) via triangle inequality and using the fact that, by Equation (70), we get the following inequality, with probability at least $1 - \frac{1}{\text{poly}(n)}$

$$\left|\left\langle \phi_{i,j}^{(h)}(y), \phi_{i',j'}^{(h)}(z)\right\rangle - \Gamma_{i,j,i',j'}^{(h)}(y,z)\right| \leq (h+1) \cdot \frac{\varepsilon}{60L^2} \cdot \frac{\sqrt{N_{i,j}^{(h)}(y) N_{i',j'}^{(h)}(z)}}{q^2}.$$

Similarly, we can prove that with probability at least $1 - \frac{1}{\text{poly}(n)}$ the following hold, simultaneously for all $i, i' \in [d_1]$ and $j, j' \in [d_2]$,

$$\left|\left\|\phi_{i,j}^{(h)}(y)\right\|_2^2 - \Gamma_{i,j,i,j}^{(h)}(y,y)\right| \leq \frac{(h+1)\varepsilon}{60L^2} \cdot \frac{N_{i,j}^{(h)}(y)}{q^2},$$

$$\left|\left\|\phi_{i',j'}^{(h)}(z)\right\|_2^2 - \Gamma_{i',j',i',j'}^{(h)}(z,z)\right| \leq \frac{(h+1)\varepsilon}{60L^2} \cdot \frac{N_{i',j'}^{(h)}(z)}{q^2}.$$

This is sufficient to prove the inductive step for statement $P_1(h)$, i.e., $\Pr[P_1(h)|P_1(h-1)] \geq 1 - \frac{1}{\text{poly}(n)}$.

Now we prove the inductive step for statement $P_2(h)$. That is, we prove that conditioned on $P_2(h-1), P_1(h)$, and $P_1(h-1), P_2(h)$ holds with probability at least $1 - \frac{1}{\text{poly}(n)}$. First note that using the definition of $\dot{\phi}^{(h)}_{i,j}(y), \dot{\phi}^{(h)}_{i',j'}(z)$ in line 7 of the algorithm and Equation (76),we find that with probability at least $1 - \frac{1}{\text{poly}(n)}$, the following holds simultaneously for all $i, i' \in [d_1]$ and $j, j' \in [d_2]$:

$$\left|\left\langle \dot{\phi}^{(h)}_{i,j}(y), \dot{\phi}^{(h)}_{i',j'}(z)\right\rangle - \frac{1}{q^2} \cdot \widetilde{\kappa}'\left(\left\langle \mu^{(h)}_{i,j}(y), \mu^{(h)}_{i',j'}(z)\right\rangle\right)\right| \leq \mathcal{O}\left(\frac{\varepsilon}{L^2}\right) \cdot \widehat{B}, \tag{80}$$

where $\widehat{B} := \frac{1}{q^2} \cdot \sqrt{\widetilde{\kappa}'\left(\|\mu^{(h)}_{i,j}(y)\|_2^2\right) \cdot \widetilde{\kappa}'\left(\|\mu^{(h)}_{i',j'}(z)\|_2^2\right)}$. By conditioning on the inductive hypothesis $P_1(h-1)$ and using Lemma 4 we have, $\left|\left\|\mu^{(h)}_{i,j}(y)\right\|_2^2 - 1\right| \leq h \cdot \frac{\varepsilon}{60L^2}$ and $\left|\left\|\mu^{(h)}_{i',j'}(z)\right\|_2^2 - 1\right| \leq h \cdot \frac{\varepsilon}{60L^2}$. Therefore, the precondition of the theorem implies that $\left|\widetilde{\kappa}'\left(\|\mu^{(h)}_{i,j}(y)\|_2^2\right) - \widetilde{\kappa}'(1)\right| \leq h \cdot \frac{\varepsilon}{20L^2}$ and $\left|\widetilde{\kappa}'\left(\|\mu^{(h)}_{i',j'}(z)\|_2^2\right) - \widetilde{\kappa}'(1)\right| \leq h \cdot \frac{\varepsilon}{20L^2}$. Consequently, because $\widetilde{\kappa}'(1) \leq 1.01\kappa'(1) = 1.01$, we find that

$$\widehat{B} \leq \frac{11}{10}\frac{1}{q^2}.$$

By plugging this into Equation (80) we get the following, with probability at least $1 - \mathcal{O}\left(\frac{\delta}{L}\right)$,

$$\left|\left\langle \dot{\phi}^{(h)}_{i,j}(y), \dot{\phi}^{(h)}_{i',j'}(z)\right\rangle - \frac{1}{q^2} \cdot \widetilde{\kappa}'\left(\left\langle \mu^{(h)}_{i,j}(y), \mu^{(h)}_{i',j'}(z)\right\rangle\right)\right| \leq \mathcal{O}\left(\frac{\varepsilon}{q^2 \cdot L^2}\right). \tag{81}$$

Furthermore, we can use the precondition of the theorem to find that Equation (75) implies the following,

$$\left|\widetilde{\kappa}'\left(\left\langle \mu^{(h)}_{i,j}(y), \mu^{(h)}_{i',j'}(z)\right\rangle\right) - \widetilde{\kappa}'(A)\right| \leq \frac{h \cdot \varepsilon}{20L^2}.$$

By incorporating the above inequality into Equation (81) using triangle inequality, we find that, with probability at least $1 - \frac{1}{\text{poly}(n)}$, the following holds simultaneously for all $i, i' \in [d_1]$ and all $j, j' \in [d_2]$:

$$\left|\left\langle \dot{\phi}^{(h)}_{i,j}(y), \dot{\phi}^{(h)}_{i',j'}(z)\right\rangle - \frac{1}{q^2} \cdot \widetilde{\kappa}'(A)\right| \leq \mathcal{O}\left(\frac{\varepsilon}{q^2 L^2}\right) + \frac{h}{q^2} \cdot \frac{\varepsilon}{20L^2}. \tag{82}$$

Since $-1 \leq A \leq 1$, we can use the precondition of the theorem to conclude $|\widetilde{\kappa}'(A) - \kappa'(A)| \leq \frac{\varepsilon}{15L^2}$. By combining this inequality with Equation (82) via triangle inequality and using the fact that $\dot{\Gamma}^{(h)}_{i,j,i',j'}(y,z) = \frac{1}{q^2} \cdot \kappa'(A)$, we get the following bound simultaneously for all $i, i' \in [d_1]$ and all $j, j' \in [d_2]$, with probability at least $1 - \frac{1}{\text{poly}(n)}$:

$$\left|\left\langle \dot{\phi}^{(h)}_{i,j}(y), \dot{\phi}^{(h)}_{i',j'}(z)\right\rangle - \dot{\Gamma}^{(h)}_{i,j,i',j'}(y,z)\right| \leq \frac{1}{q^2} \cdot \frac{\varepsilon}{8L}. \tag{83}$$

Similarly we can prove that with probability at least $1 - \frac{1}{\text{poly}(n)}$, the following hold simultaneously for all $i, i' \in [d_1]$ and all $j, j' \in [d_2]$,

$$\left|\left\|\dot{\phi}^{(h)}_{i,j}(y)\right\|_2^2 - \dot{\Gamma}^{(h)}_{i,j,i,j}(y,y)\right| \leq \frac{1}{q^2} \cdot \frac{\varepsilon}{8L}, \text{ and } \left|\left\|\dot{\phi}^{(h)}_{i',j'}(z)\right\|_2^2 - \dot{\Gamma}^{(h)}_{i',j',i',j'}(z,z)\right| \leq \frac{1}{q^2} \cdot \frac{\varepsilon}{8L}. \tag{84}$$

We will use Equation (83) and Equation (84) to prove the inductive step for $P_2(h)$.

Next, we consider two cases for the value of h. When $h < L$, the vectors $\psi^{(h)}_{i,j}(y), \psi^{(h)}_{i',j'}(z)$ are defined in line 10 and when $h = L$, these vectors are defined differently in line 12. First we consider the case of $h < L$. If we let $f_{i,j} := \psi^{(h-1)}_{i,j}(y) \otimes \dot{\phi}^{(h)}_{i,j}(y)$ and $g_{i',j'} := \psi^{(h-1)}_{i',j'}(z) \otimes \dot{\phi}^{(h)}_{i',j'}(z)$ and $\eta^{(h)}_{i,j}(y) := \left(Q^2 \cdot f_{i,j}\right) \oplus \phi^{(h)}_{i,j}(y)$ and $\eta^{(h)}_{i',j'}(z) := \left(Q^2 \cdot g_{i',j'}\right) \oplus \phi^{(h)}_{i',j'}(z)$, then by Theorem 5 and union bound, with probability at least $1 - \frac{1}{\text{poly}(n)}$, we have the following inequalities simultaneously

for all $i, i' \in [d_1]$ and $j, j' \in [d_2]$:

$$\begin{aligned}
&\left|\left\langle \eta_{i,j}^{(h)}(y), \eta_{i',j'}^{(h)}(z)\right\rangle - \langle f_{i,j}, g_{i',j'}\rangle - \left\langle \phi_{i,j}^{(h)}(y), \phi_{i',j'}^{(h)}(z)\right\rangle\right| \leq \mathcal{O}\left(\frac{\varepsilon}{L}\right) \cdot \|f_{i,j}\|_2 \|g_{i',j'}\|_2 \\
&\left\|\eta_{i,j}^{(h)}(y)\right\|_2^2 \leq \frac{11}{10} \cdot \|f_{i,j}\|_2^2 + \left\|\phi_{i,j}^{(h)}(y)\right\|_2^2 \\
&\left\|\eta_{i',j'}^{(h)}(z)\right\|_2^2 \leq \frac{11}{10} \cdot \|g_{i',j'}\|_2^2 + \left\|\phi_{i',j'}^{(h)}(z)\right\|_2^2
\end{aligned} \tag{85}$$

Now we bound the term $\left|\left\langle \eta_{i,j}^{(h)}(y), \eta_{i',j'}^{(h)}(z)\right\rangle - \langle f_{i,j}, g_{i',j'}\rangle - \left\langle \phi_{i,j}^{(h)}(y), \phi_{i',j'}^{(h)}(z)\right\rangle\right|$ using Equation (85), Equation (84), and Lemma 4 along with inductive hypotheses $P_2(h-1)$. With probability at least $1 - \frac{1}{\text{poly}(n)}$ the following holds simultaneously for all $i, i' \in [d_1]$ and all $j, j' \in [d_2]$:

$$\begin{aligned}
&\left|\left\langle \eta_{i,j}^{(h)}(y), \eta_{i',j'}^{(h)}(z)\right\rangle - \langle f_{i,j}, g_{i',j'}\rangle - \left\langle \phi_{i,j}^{(h)}(y), \phi_{i',j'}^{(h)}(z)\right\rangle\right| \\
&\quad \leq \mathcal{O}\left(\frac{\varepsilon}{L}\right) \cdot \sqrt{\Pi_{i,j,i,j}^{(h-1)}(y,y) \cdot \dot{\Gamma}_{i,j,i,j}^{(h)}(y,y) \cdot \Pi_{i',j',i',j'}^{(h-1)}(z,z) \cdot \dot{\Gamma}_{i',j',i',j'}^{(h)}(z,z)} \\
&\quad = \mathcal{O}\left(\frac{\varepsilon \cdot h}{L}\right) \cdot \frac{\sqrt{N_{i,j}^{(h)}(y) \cdot N_{i',j'}^{(h)}(z)}}{q^2},
\end{aligned}$$

where the last line above follows from Lemma 4 together with the fact that $\dot{\Gamma}_{i,j,i,j}^{(h)}(y,y) = \dot{\Gamma}_{i',j',i',j'}^{(h)}(z,z) = \frac{1}{q^2}$. By combining the above with inductive hypotheses $P_1(h), P_2(h-1)$ and Equation (83) via triangle inequality and invoking Lemma 4 we get that the following holds simultaneously for all $i, i' \in [d_1]$ and all $j, j' \in [d_2]$, with probability at least $1 - \frac{1}{\text{poly}(n)}$,

$$\begin{aligned}
&\left|\left\langle \eta_{i,j}^{(h)}(y), \eta_{i',j'}^{(h)}(z)\right\rangle - \Pi_{i,j,i',j'}^{(h-1)}(y,z) \cdot \dot{\Gamma}_{i,j,i',j'}^{(h)}(y,z) - \Gamma_{i,j,i',j'}^{(h)}(y,z)\right| \\
&\leq \frac{\varepsilon}{10} \cdot \frac{(h-1)^2}{L+1} \cdot \sqrt{N_{i,j}^{(h)}(y) \cdot N_{i',j'}^{(h)}(z)} \cdot \left(\left|\dot{\Gamma}_{i,j,i',j'}^{(h)}(y,z)\right| + \frac{1}{q^2} \cdot \frac{\varepsilon}{8L}\right) + \frac{1}{q^2} \cdot \frac{\varepsilon}{8L} \cdot \left|\Pi_{i,j,i',j'}^{(h-1)}(y,z)\right| \\
&+ \frac{(h+1) \cdot \varepsilon}{60L^2} \cdot \frac{\sqrt{N_{i,j}^{(h)}(y) \cdot N_{i',j'}^{(h)}(z)}}{q^2} + \mathcal{O}\left(\frac{\varepsilon \cdot h}{L}\right) \cdot \frac{\sqrt{N_{i,j}^{(h)}(y) \cdot N_{i',j'}^{(h)}(z)}}{q^2} \\
&\leq \frac{\varepsilon}{10} \cdot \frac{(h-1)^2}{L+1} \cdot \frac{\sqrt{N_{i,j}^{(h)}(y) \cdot N_{i',j'}^{(h)}(z)}}{q^2} \cdot \left(1 + \frac{\varepsilon}{8L}\right) + \frac{h-1}{q^2} \cdot \frac{\varepsilon}{8L} \cdot \sqrt{N_{i,j}^{(h)}(y) \cdot N_{i',j'}^{(h)}(z)} \\
&+ \left(\frac{(h+1) \cdot \varepsilon}{60L^2} + \mathcal{O}\left(\frac{\varepsilon \cdot h}{L}\right)\right) \cdot \frac{\sqrt{N_{i,j}^{(h)}(y) \cdot N_{i',j'}^{(h)}(z)}}{q^2} \\
&\leq \frac{\varepsilon}{10} \cdot \frac{h^2 - h/2}{L+1} \cdot \frac{\sqrt{N_{i,j}^{(h)}(y) \cdot N_{i',j'}^{(h)}(z)}}{q^2}.
\end{aligned}$$

By plugging the above bound into the definition of in line 10 of the algorithm using triangle inequality and using Equation (72) we get the following with probability at least $1 - \frac{1}{\text{poly}(n)}$:

$$\begin{aligned}
&\left|\left\langle \psi_{i,j}^{(h)}(y), \psi_{i',j'}^{(h)}(z)\right\rangle - \Pi_{i,j,i',j'}^{(h)}(y,z)\right| \\
&\leq \frac{\varepsilon}{10} \cdot \frac{h^2 - h/2}{L+1} \cdot \sum_{a=-\frac{q-1}{2}}^{\frac{q-1}{2}} \sum_{b=-\frac{q-1}{2}}^{\frac{q-1}{2}} \frac{\sqrt{N_{i+a,j+b}^{(h)}(y) \cdot N_{i'+a,j'+b}^{(h)}(z)}}{q^2} \\
&\leq \frac{\varepsilon}{10} \cdot \frac{h^2 - h/2}{L+1} \cdot \sqrt{\sum_{a=-\frac{q-1}{2}}^{\frac{q-1}{2}} \sum_{b=-\frac{q-1}{2}}^{\frac{q-1}{2}} \frac{N_{i+a,j+b}^{(h)}(y)}{q^2}} \cdot \sqrt{\sum_{a=-\frac{q-1}{2}}^{\frac{q-1}{2}} \sum_{b=-\frac{q-1}{2}}^{\frac{q-1}{2}} \frac{N_{i'+a,j'+b}^{(h)}(z)}{q^2}} \\
&\leq \frac{\varepsilon}{10} \cdot \frac{h^2}{L+1} \cdot \sqrt{N_{i,j}^{(h+1)}(y) \cdot N_{i',j'}^{(h+1)}(z)}.
\end{aligned} \tag{86}$$

Similarly, we can prove that with probability at least $1 - \frac{1}{\text{poly}(n)}$ the following hold simultaneously for all $i, i' \in [d_1]$ and all $j, j' \in [d_2]$,

$$\left| \left\| \psi_{i,j}^{(h)}(y) \right\|_2^2 - \Pi_{i,j,i,j}^{(h)}(y,y) \right| \leq \frac{\varepsilon}{10} \cdot \frac{h^2}{L+1} \cdot N_{i,j}^{(h+1)}(y),$$
$$\left| \left\| \psi_{i',j'}^{(h)}(z) \right\|_2^2 - \Pi_{i',j',i',j'}^{(h)}(z,z) \right| \leq \frac{\varepsilon}{10} \cdot \frac{h^2}{L+1} \cdot N_{i',j'}^{(h+1)}(z).$$

This is sufficient to prove the inductive step for statement $P_2(h)$, in the case of $h < L$, i.e., $\Pr[P_2(h)|P_2(h-1), P_1(h), P_1(h-1)] \geq 1 - \frac{1}{\text{poly}(n)}$.

Now we prove the inductive step for $P_2(h)$ in the case of $h = L$. Similar to before, if we let $f_{i,j} := \psi_{i,j}^{(L-1)}(y) \otimes \dot{\phi}_{i,j}^{(L)}(y)$ and $g_{i',j'} := \psi_{i',j'}^{(L-1)}(z) \otimes \dot{\phi}_{i',j'}^{(L)}(z)$, then by (12), we have $\psi_{i,j}^{(L)}(y) = \left(Q^2 \cdot f_{i,j}\right)$ and $\psi_{i',j'}^{(L)}(z) = \left(Q^2 \cdot g_{i',j'}\right)$. Thus by Theorem 5 and union bound, we find that, with probability at least $1 - \frac{1}{\text{poly}(n)}$, the following inequality holds simultaneously for all $i, i' \in [d_1]$ and $j, j' \in [d_2]$:

$$\left| \left\langle \psi_{i,j}^{(L)}(y), \psi_{i',j'}^{(L)}(z) \right\rangle - \langle f_{i,j}, g_{i',j'} \rangle \right| \leq \mathcal{O}\left(\frac{\varepsilon}{L}\right) \cdot \|f_{i,j}\|_2 \|g_{i',j'}\|_2 .$$

Therefore, using (84) and Lemma 4 along with inductive hypotheses $P_2(L-1)$, with probability at least $1 - \frac{1}{\text{poly}(n)}$, the following holds simultaneously for all $i, i' \in [d_1]$ and $j, j' \in [d_2]$,

$$\begin{aligned}
&\left| \left\langle \psi_{i,j}^{(L)}(y), \psi_{i',j'}^{(L)}(z) \right\rangle - \langle f_{i,j}, g_{i',j'} \rangle \right| \\
&\quad \leq \mathcal{O}\left(\frac{\varepsilon}{L}\right) \cdot \sqrt{\Pi_{i,j,i,j}^{(L-1)}(y,y) \cdot \dot{\Gamma}_{i,j,i,j}^{(L)}(y,y) \cdot \Pi_{i',j',i',j'}^{(L-1)}(z,z) \cdot \dot{\Gamma}_{i',j',i',j'}^{(L)}(z,z)} \\
&\quad = \mathcal{O}(\varepsilon) \cdot \frac{\sqrt{N_{i,j}^{(L)}(y) \cdot N_{i',j'}^{(L)}(z)}}{q^2}.
\end{aligned}$$

By combining the above with inductive hypotheses $P_1(L), P_2(L-1)$ and Equation (83) via triangle inequality and invoking Lemma 4 and also using the definition of $\Pi^{(L)}(y,z)$ given in Equation (73), we get that the following holds, simultaneously for all $i, i' \in [d_1]$ and $j, j' \in [d_2]$, with probability at least $1 - \frac{1}{\text{poly}(n)}$,

$$\begin{aligned}
&\left| \left\langle \psi_{i,j}^{(L)}(y), \psi_{i',j'}^{(L)}(z) \right\rangle - \Pi_{i,j,i',j'}^{(L)}(y,z) \right| \\
&\leq \frac{\varepsilon}{10} \cdot \frac{(L-1)^2}{L+1} \cdot \sqrt{N_{i,j}^{(L)}(y) \cdot N_{i',j'}^{(L)}(z)} \cdot \left(\left| \dot{\Gamma}_{i,j,i',j'}^{(L)}(y,z) \right| + \frac{1}{q^2} \cdot \frac{\varepsilon}{8L} \right) + \frac{1}{q^2} \cdot \frac{\varepsilon}{8L} \cdot \left| \Pi_{i,j,i',j'}^{(L-1)}(y,z) \right| \\
&+ \frac{(L+1) \cdot \varepsilon}{60L^2} \cdot \frac{\sqrt{N_{i,j}^{(L)}(y) \cdot N_{i',j'}^{(L)}(z)}}{q^2} + \mathcal{O}(\varepsilon) \cdot \frac{\sqrt{N_{i,j}^{(L)}(y) \cdot N_{i',j'}^{(L)}(z)}}{q^2} \\
&\leq \frac{\varepsilon}{10} \cdot \frac{(L-1)^2}{L+1} \cdot \frac{\sqrt{N_{i,j}^{(L)}(y) \cdot N_{i',j'}^{(L)}(z)}}{q^2} \cdot \left(1 + \frac{\varepsilon}{8L}\right) + \frac{\varepsilon}{8q^2} \cdot \sqrt{N_{i,j}^{(L)}(y) \cdot N_{i',j'}^{(L)}(z)} \\
&+ \left(\frac{(L+1) \cdot \varepsilon}{60L^2} + \mathcal{O}(\varepsilon) \right) \cdot \frac{\sqrt{N_{i,j}^{(L)}(y) \cdot N_{i',j'}^{(L)}(z)}}{q^2} \\
&\leq \frac{\varepsilon \cdot (L-1)}{10} \cdot \frac{\sqrt{N_{i,j}^{(L)}(y) \cdot N_{i',j'}^{(L)}(z)}}{q^2}.
\end{aligned}$$

This proves the inductive step for statement $P_2(h)$, in the case of $h = L$, i.e., $\Pr[P_2(L)|P_2(L-1), P_1(L), P_1(L-1)] \geq 1 - \frac{1}{\text{poly}(n)}$. The induction is complete and hence the correctness of Algorithm 4 is proved by union bounding over all $h = 0, 1, 2, \ldots L$.

The runtime of the algorithm immediately follows by invoking Theorem 5 because computing vector $Z_{i,j,\ell}^{(h)}(x)$ for every i, j, ℓ and $h = 1, 2, \ldots L$ dominates the runtime of this algorithm. □

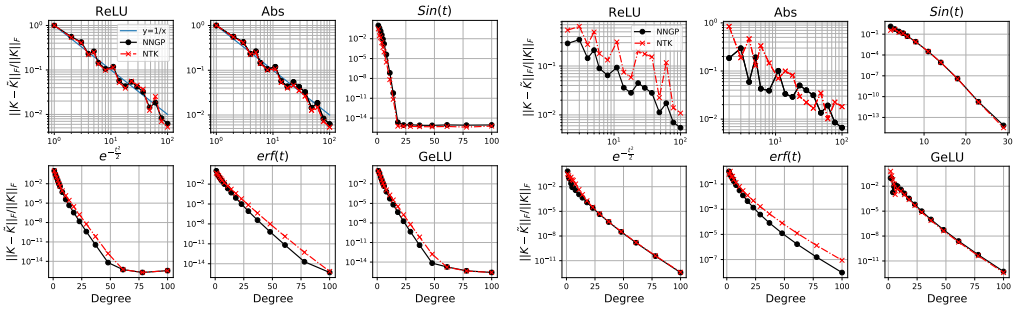

Figure 3: Relative errors of dual kernel approximation via the Hermite polynomial approximation under synthetic dataset with **(left)** $n = 1{,}000, d = 256$ FC1, **(right)** $n = 10, d = 32 \times 32 \times 3$ Myrtle-5.

As an example, let us invoke Algorithm 4 and Theorem 6 on the CNTK with GAP corresponding to the normalized Gaussian dual kernel K_G, defined per Equation (13). Note that the dot-product factor corresponding to this dual kernel is $\kappa(t) = \exp(t-1)$. The truncated Taylor series of this function is $\widetilde{\kappa}(t) = \sum_{j=0}^{p} \frac{t^j}{e \cdot j!}$ and the truncated Taylor series expansion of the derivative of this function is $\widetilde{\kappa}'(t) = \sum_{j=0}^{p} \frac{t^j}{e \cdot j!}$. If $p = \Omega(\log n)$ then it can be verified that polynomials $\widetilde{\kappa}(t), \widetilde{\kappa}'(t)$ satisfy the preconditions of Theorem 6. Therefore, by Theorem 6, we can sketch the CNTK kernel using $\mathcal{O}\left(\frac{L^4}{\varepsilon^2} \cdot d_1 d_2 \cdot \text{poly}\left(\log n\right)\right)$ running time. Also the target dimension of the sketch is $m' = \mathcal{O}\left(\frac{L^2}{\varepsilon^2} \log^3 n\right)$. So the runtime of our Algorithm 4 is only linear in the number of image pixels $d_1 d_2$, which is in stark contrast to quadratic scaling of the exact CNTK computation [9]. In fact, using our CNTK sketching method, the kernel regression can be solved approximately in time $\mathcal{O}\left(\frac{L^4}{\varepsilon^2} \cdot d_1 d_2 \cdot n \text{ poly}\left(\log n\right) + m'^2 \cdot n\right) = \mathcal{O}\left(\left(\frac{L^4}{\varepsilon^2} \cdot d_1 d_2 + \frac{L^4}{\varepsilon^4}\right) \cdot n \cdot \text{poly}\left(\log n\right)\right)$, which is significantly faster than the exact kernel regression which takes $\Omega\left(L(d_1 d_2 \cdot n)^2\right)$ when the number of pixels $d_1 d_2$ or the training set size n are large.

E Gauss-Hermite Quadrature Derivation

Here we provide more details on Section 3.3. Utilizing the whitening transformation of covariance $\mathbf{\Lambda}$ used in the proof of Theorem 3 in Appendix B.5 the dual activation function can be expressed as

$$k_\sigma(a,b,c) := \mathop{\mathbb{E}}_{(u,v)\sim\mathcal{N}(0,\mathbf{\Lambda})} \left[\sigma(u)\sigma(v)\right] \tag{87}$$

$$= \mathop{\mathbb{E}}_{(\alpha,\beta)\sim\mathcal{N}(0,\boldsymbol{I}_2)} \left[\sigma(a\alpha) \cdot \sigma(bc\alpha + b\sqrt{1-c^2}\beta)\right] \tag{88}$$

$$= \frac{1}{2\pi} \int\int d\alpha d\beta e^{-\frac{\alpha^2}{2}} e^{-\frac{\beta^2}{2}} \left[\sigma(a\alpha) \cdot \sigma(bc\alpha + b\sqrt{1-c^2}\beta)\right] \tag{89}$$

$$= \frac{1}{\pi} \int\int d\alpha d\beta e^{-\alpha^2} e^{-\beta^2} \left[\sigma(\sqrt{2}a\alpha) \cdot \sigma(\sqrt{2}bc\alpha + \sqrt{2}b\sqrt{1-c^2}\beta)\right] \tag{90}$$

$$\approx \frac{1}{\pi} \sum_{i=1}^{q} \sum_{j=1}^{q} w_i w_j \left[\sigma(\sqrt{2}ax_i) \cdot \sigma(\sqrt{2}bcx_i + \sqrt{2}b\sqrt{1-c^2}x_j)\right]. \tag{91}$$

Here (x_i, w_i), correspond to roots of q-th degree (Physicist's) Hermite polynomial $H_q(x)$ and associated weights [55]

$$w_i = \frac{2^{q-1} q! \sqrt{\pi}}{q^2 \left(H_{q-1}(x_i)\right)^2} = \frac{q! \sqrt{\pi}}{q^2 (h_{q-1}(\sqrt{2}x_i))^2} \tag{92}$$

where the conversion between physicist's to probabilist's convention $H_n(x) = 2^{\frac{n}{2}} h_n(\sqrt{2}x)$. The roots are obtained by Golub-Welsch algorithm [63] and can be found in scientific computing package

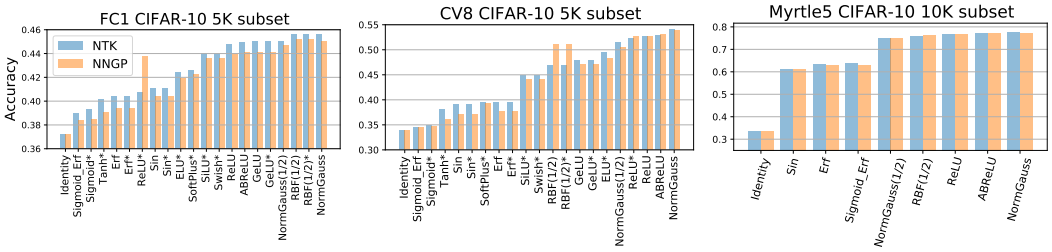

Figure 4: **Classification performance on a CIFAR-10 subset of various architectures.** We compare performance of various activation functions in neural kernels. $*$ denotes that Hermite-quadrature was used to numerically compute the dual activation functions. The slopes of ABReLU are chosen to match the Normalized Gaussian.

such as Scipy [64]'s `scipy.special.roots_hermite` function. For alternative parameterization for multivariate Gauss-Hermite quadrature, refer to notes by Jäckel [65].

For activation function where exact dual activation is known, one can measure the error from the quadrture. In Figure 3, we compute errors for ReLU, Abs (i.e., $\sigma(t) = |t|$), $\sin$, Gaussian, erf and GeLU activations. For non-smooth activation (ReLU, Abs), approximation error decays as power-law like where as for smooth activation the error decays exponentially as one increases Hermite polynomial degree q.

We utilize this method as well as our expanded dual activation Table 1 to compare performance of various activation functions on CIFAR-10 dataset. In Figure 4, we study three architectures; 1 hidden layer fully connected network (FC1, equivalent to pure dual activation kernel), 8 layer convolutional network with vectorization (CV8), and Myrtle5 network. We compared classification performance on subset of CIFAR-10. In each plot activation function is sorted by NTK's classification performance. One notable observation is that normalized Gaussian shows consistently best performance across architecture. Also note that smooth activations computed with Gauss-Hermite quadrature (denoted by $*$) shows almost identical performance when analytic form is available (e.g. GeLU, Erf, RBF(1/2)). Notable outlier is FC1 NTK with ReLU, however we expect that non-smooth activation may be approximated poorly. It's also interesting to observer sigmoid-like activations (Sigmoid, Tanh, Erf) performs poorly across the board whereas ReLU-like activations (Normalized Gaussian, ABReLU, ReLU, GeLU, RBF) are among high performant group.

This is implemented as `ElementwiseNumerical` in our code.

F Table of dual activation functions

We describe dual kernel functions of several activations and their derivatives in Table 2. One can generalize duel kernels of affine transformations of these activations. Specifically, if $\widetilde{\sigma}(t) = A \cdot \sigma(Bt) + C$ for some $A, B, C \in \mathbb{R}$ then

$$k_{\widetilde{\sigma}}(a, b, c) = A^2 \cdot k_\sigma(Ba, Bb, c) + C^2 + A\,C \mathop{\mathbb{E}}_{t \sim \mathcal{N}(0,1)} [\sigma(Bat) + \sigma(Bbt)] \tag{93}$$

which follows from that

$$\begin{aligned} k_{\widetilde{\sigma}}(a, b, c) &= \mathop{\mathbb{E}}_{(u,v) \sim \mathcal{N}(0, \mathbf{\Lambda}_{a,b,c})} \left[A^2 \sigma(Bu)\sigma(Bv) + C^2 + AC\left(\sigma(Bu) + \sigma(Bv)\right) \right] \\ &= A^2 \mathop{\mathbb{E}}_{(u,v) \sim \mathcal{N}(0, \mathbf{\Lambda}_{a,b,c})} [\sigma(Bu)\sigma(Bv)] + C^2 + AC \mathop{\mathbb{E}}_{(u,v) \sim \mathcal{N}(0, \mathbf{\Lambda}_{a,b,c})} [\sigma(Bu) + \sigma(Bv)] \\ &= A^2 \cdot k_\sigma(Ba, Bb, c) + C^2 + AC \left(\mathop{\mathbb{E}}_{u \sim \mathcal{N}(0, a^2)} [\sigma(Bu)] + \mathop{\mathbb{E}}_{u \sim \mathcal{N}(0, b^2)} [\sigma(Bv)] \right) \\ &= A^2 \cdot k_\sigma(Ba, Bb, c) + C^2 + AC \left(\mathop{\mathbb{E}}_{t \sim \mathcal{N}(0,1)} [\sigma(Bat)] + \mathop{\mathbb{E}}_{t \sim \mathcal{N}(0,1)} [\sigma(Bbt)] \right). \end{aligned} \tag{94}$$

Below we provide detailed expressions of omitted dual kernel formulations in the table.

Activation	$\sigma(t)$	Dual kernel $k_\sigma(a,b,c)$	$k_{\dot\sigma}(a,b,c)$	Implemented as
Rectified monomials [44]	$t^n \cdot \mathbb{1}_{\{t\geq 0\}}$	$\frac{(ab)^n}{2\pi} J_n\left(\cos^{-1}(c)\right)$	$\frac{n^2(ab)^{n-1}}{2\pi} J_{n-1}\left(\cos^{-1}(c)\right)$	`RectifiedMonomial`
ReLU [44]	$\max(t,0)$	$\frac{ab}{2\pi}\left(\sqrt{1-c^2}+(\pi-\cos^{-1}(c))c\right)$	$\frac{1}{2\pi}(\pi-\cos^{-1}(c))$	`ReLU`
ABReLU [42, 50, 51]	$A\max(t,0)$ $+B\max(-t,0)$	$\frac{ab(B-A)^2\left(\sqrt{1-c^2}+\left(\pi-\cos^{-1}(c)\right)c\right)}{2\pi}$ $+ABabc$	Appendix F.2	`ABReLU`
Sinusoidal [37, 38]	$\sin(t)$	$e^{-\frac{a^2+b^2}{2}}\sinh(abc)$	$e^{-\frac{a^2+b^2}{2}}\cosh(abc)$	`Sin`
	$\cos(t)$	$e^{-\frac{a^2+b^2}{2}}\cosh(abc)$	$e^{-\frac{a^2+b^2}{2}}\sinh(abc)$	
	$A\sin(Bt+C)$	Equation (111)	Equation (112)	
Error function [5, 43]	$\mathrm{erf}(t)$	$\frac{2}{\pi}\sin^{-1}\left(\frac{2abc}{\sqrt{(1+2a^2)(1+2b^2)}}\right)$	$\frac{4}{\pi}\frac{1}{\sqrt{(1+2a^2)(1+2b^2)-4(abc)^2}}$	`Erf`
Gaussian [43]	$\exp(-At^2)$	$\frac{1}{\sqrt{(2Aa^2+1)(2Ab^2+1)-(2Aabc)^2}}$	$\frac{4A^2abc}{((2Aa^2+1)(2Ab^2+1)-(2Aabc)^2)^{3/2}}$	`Gaussian`
Exponential [46, 52]	$\exp(At)$	$\exp\left(\frac{A^2}{2}\left(a^2+b^2+2abc\right)\right)$	$A^2\exp\left(\frac{A^2}{2}\left(a^2+b^2+2abc\right)\right)$	`Exp`
GeLU [48]	$\frac{t}{2}\left(1+\mathrm{erf}\left(\frac{t}{\sqrt{2}}\right)\right)$	Equation (125)	Appendix F.6	`Gelu`
Gabor	$\exp(-t^2)\sin(t)$	Equation (137)	Appendix F.8	`Gabor`
Polynomial	$\sum_j c_j t^j$	Theorem 1	Theorem 1	`Polynomial`
Normalized Gaussian [54]	Unknown	$ab\exp(c-1)$	$\exp(c-1)$	`ExpNormalized`
RBF [45]	$\sqrt{2}\sin\left(\sqrt{2A}t+\frac{\pi}{4}\right)$	$\exp\left(-A\left(a^2+b^2-2abc\right)\right)$	$2A\exp\left(-A\left(a^2+b^2-2abc\right)\right)$	`Rbf`

Table 2: Dual kernels of activation and its derivative for various functions.

F.1 Rectified monomials

Cho and Saul [44] proposed closed-form expressions of dual kernel functions for rectified activations, i.e., $\sigma(t) = t^n \cdot \mathbb{1}_{\{t\geq 0\}}$ for $n \geq 0$, as

$$k_\sigma(a,b,c) = \frac{(ab)^n}{2\pi} \cdot J_n\left(\cos^{-1}(c)\right) \tag{95}$$

where for $\theta = \cos^{-1}(c) \in [0,\pi]$

$$J_n(\theta) := (-1)^n (\sin\theta)^{(2n+1)} \left(\frac{1}{\sin\theta}\frac{\partial}{\partial\theta}\right)^n \left(\frac{\pi-\theta}{\sin\theta}\right). \tag{96}$$

For $n = 0$ and 1

$$J_0(\theta) = \pi - \theta, \qquad J_1(\theta) = \sin\theta + (\pi-\theta)\cos\theta. \tag{97}$$

Applying Theorem 3 provides that

$$k_{\sigma'}(a,b,c) = \frac{n^2(ab)^{n-1}}{2\pi} \cdot J_{n-1}\left(\cos^{-1}(c)\right). \tag{98}$$

These are implemented in our code as `RectifiedMonomial` (with a special case of `Sign` for convenience).

F.2 ABReLU, Leaky ReLU, Abs

ABReLU activation function is given by

$$\sigma(t) = A\min(t,0) + B\max(t,0), \quad \text{for } A, B \in \mathbb{R} \tag{99}$$

The dual kernel functions can be obtained by extension of [44] which is worked out in [50, 51]

$$k_\sigma(a,b,c) = ab\left(\frac{(B-A)^2}{2\pi}J_1\left(\cos^{-1}(c)\right) + ABc\right) \tag{100}$$

$$= ab\left(\frac{(B-A)^2}{2\pi}\left(\sqrt{1-c^2} + (\pi - \cos^{-1}(c))c\right) + ABc\right) \tag{101}$$

and

$$k_{\sigma'}(a,b,c) = \frac{(B-A)^2}{2\pi}J_0(\cos^{-1}(c) + AB \tag{102}$$

$$= \frac{(B-A)^2}{2\pi}\left(\pi - \cos^{-1}(c)\right) + AB. \tag{103}$$

A special case of ABReLU covers leaky ReLU [66] ($B = 1$), that is,

$$\sigma(t) = A\min(t,0) + \max(t,0)\,, \tag{104}$$

and the corresponding dual kernel functions are

$$k_\sigma(a,b,c) = ab\left(\frac{(1-A)^2}{2\pi}J_1\left(\cos^{-1}(c)\right) + Ac\right), \tag{105}$$

and

$$k_{\sigma'}(a,b,c) = \frac{(1-A)^2}{2\pi}J_0\left(\cos^{-1}(c)\right) + A. \tag{106}$$

Another special case is the absolute value function (Abs) ($A = -1, B = 1$), that is,

$$\sigma(t) = |t|\,, \tag{107}$$

and the corresponding dual kernel functions are

$$k_\sigma(a,b,c) = ab\left(\frac{2}{\pi}J_1\left(\cos^{-1}(c)\right) - c\right) \tag{108}$$

and

$$k_{\sigma'}(a,b,c) = 1 - \frac{2}{\pi}\cos^{-1}(c)\,. \tag{109}$$

These are respectively implemented as `ABRelu`, `LeakyRelu`, and `Abs` in [42].

F.3 Sinusoidal and RBF

A generalized sinusoidal activation is given by

$$\sigma(t) = A\sin(Bt + C). \tag{110}$$

The corresponding dual kernels are

$$k_\sigma(a,b,c) = \frac{A^2}{2}\cdot e^{-\frac{B^2(a^2+b^2)}{2}}\left(e^{abcB^2} - \cos(2C)e^{-abcB^2}\right) \tag{111}$$

$$k_{\sigma'}(a,b,c) = \frac{A^2B^2}{2}\cdot e^{-\frac{B^2(a^2+b^2)}{2}}\left(e^{abcB^2} + \cos(2C)e^{-abcB^2}\right). \tag{112}$$

Note that the generalized sinusoidal activation with $a = \sqrt{2}$, $b = \sqrt{2A}$, and $c = \frac{\pi}{4}$ gives that

$$k_\sigma(a,b,c) = \exp\left(-A\left(a^2 + b^2 - 2abc\right)\right), \tag{113}$$

$$k_{\sigma'}(a,b,c) = 2A\exp\left(-A\left(a^2 + b^2 - 2abc\right)\right)\,, \tag{114}$$

which corresponds to (translation invariant) the Gaussian RBF kernel:

$$k_{\text{RBF}}(x,y) = \exp\left(-dA\left\|x-y\right\|^2\right)\,. \tag{115}$$

for some $x, y \in \mathbb{R}^d$.

Moreover, one could consider mixture of activation functions as discussed in Louart et al. [47], Adlam et al. [67] of 50% cos and 50% sin which also leads to stationary kernel

$$k_{\cos+\sin}(a,b,c) = \frac{1}{2}\exp(-\frac{1}{2}(a^2 + b^2 - 2abc)). \tag{116}$$

In order to obtain stationary kernel with respect to inputs, one only needs to insert these transformation at the first layer of the network as highlighted in implicit neural representation (e.g. NeRF) [37, 38].

These are implemented in our code as `Sin`, `Cos`, and `Rbf`.

F.4 Error function

The error function is given by

$$\sigma(t) = \frac{2}{\sqrt{\pi}} \int_0^t e^{-x^2} dx. \tag{117}$$

Following [43] and applying Theorem 3, we get

$$k_\sigma(a,b,c) = \frac{2}{\pi} \sin^{-1}\left(\frac{2abc}{\sqrt{(1+a^2)(1+b^2)}}\right), \tag{118}$$

$$k_{\sigma'}(a,b,c) = \frac{4}{\pi} \frac{1}{\sqrt{(1+2a^2)(1+2b^2) - 4(abc)^2}}. \tag{119}$$

An affine transformation of the error function could behave similar to sigmoid activation function with range $(0,1)$, that is,

$$\sigma_{\text{sigmoid-like}}(x) = \frac{1}{2}\left(\operatorname{erf}\left(\frac{x}{2.4020563531719796}\right) + 1\right). \tag{120}$$

The corresponding dual kernels can be obtained by applying affine transformation to that of error function as discussed in Equation (93). The error function is implemented in [42] as `Erf`, and we release `Sigmoid_like` in our code.

F.5 Gaussian function

Consider Gaussian function

$$\sigma(t) = \exp(-At^2).$$

One can obtain k_σ [43],

$$k_\sigma(a,b,c) = \frac{1}{\sqrt{(2Aa^2+1)(2Ab^2+1) - (2Aabc)^2}} \tag{121}$$

and using Theorem 3 obtain

$$k_{\sigma'}(a,b,c) = \frac{4A^2abc}{((2Aa^2+1)(2Ab^2+1) - (2Aabc)^2)^{3/2}}. \tag{122}$$

Note that Gaussian function itself can be obtained as derivative of Affine Erf thus could use Theorem 3 with Affine Erf. This function is implemented as `Gaussian` in our code.

F.6 GeLU

The Gaussian Error Linear Unit (GeLU) [33] is defined as

$$\sigma(t) = \frac{t}{2}\left(1 + \operatorname{erf}\left(\frac{t}{\sqrt{2}}\right)\right) = \frac{t}{\sqrt{2\pi}} \int_{-\infty}^t e^{-\frac{s^2}{2}} ds, \tag{123}$$

where $\operatorname{erf}(\cdot)$ is the Gauss error function. For efficiency, sometimes approximate formulation

$$\tilde{\sigma}(t) = \frac{t}{2}\left(1 + \tanh\left(\sqrt{\frac{2}{\pi}}(t + 0.044715t^3)\right)\right), \tag{124}$$

is used. We note that GeLU activation function is becoming popular in recent language models such as BERT [34], ALBERT [68], RoBERTa [69] and GPT [35, 36]. The corresponding dual kernel is studied in Tsuchida et al. [48]:

$$k_{\text{GeLU}}(a,b,c) = \frac{abc}{4} + \frac{a^2b^2}{2\pi}\left(\frac{c^2+1+a^2+b^2+a^2b^2(1-c^2)}{(1+a^2)(1+b^2)\sqrt{1+a^2+b^2+a^2b^2(1-c^2)}} + \frac{c}{ab}\tan^{-1}\left(\frac{abc}{\sqrt{1+a^2+b^2+a^2b^2(1-c^2)}}\right)\right). \tag{125}$$

Using Theorem 3, we have

$$k_{\mathrm{GeLU}'}(a,b,c) = \frac{1}{4} + \frac{\left(2-a^2b^2\right)abc(1+a^2)(1+b^2) + \left(a^2b^2-1\right)(abc)^3}{2\pi(1+a^2)(1+b^2)\left(1+a^2+b^2+a^2b^2(1-c^2)\right)^{3/2}} + \frac{1}{2\pi}\tan^{-1}\left(\frac{abc}{\sqrt{1+a^2+b^2+a^2b^2(1-c^2)}}\right) + \frac{abc}{2\pi}\frac{1}{\sqrt{1+a^2+b^2+a^2b^2(1-c^2)}}. \tag{126}$$

This is implemented in our code as `Gelu`.

F.7 Monomials

Consider monomials

$$\sigma_n(t) = t^n, \quad n \in \mathbb{N}. \tag{127}$$

The dual activation function is given in terms of Hypergeometric function ${}_2F_1$. For even power $n \in 2\mathbb{Z}$

$$k_{\sigma_n}(a,b,c) = \frac{(2ab)^n\left(1-c^2\right)^{n/2}}{\pi}\Gamma\left(\frac{n+1}{2}\right)^2 {}_2F_1\left(-\frac{n}{2},\frac{n+1}{2};\frac{1}{2};\frac{c^2}{c^2-1}\right) \tag{128}$$

For odd power $n \in 2\mathbb{Z}+1$

$$k_{\sigma_n}(a,b,c) = \frac{2^n(ab)^{n+1}\left(1-c^2\right)^{\frac{n-1}{2}}\Gamma\left(\frac{n}{2}+1\right)\Gamma\left(\frac{n}{2}\right)}{\pi(n+1)c}\left(2c^2\,{}_2F_1\left(\frac{1}{2}-\frac{n}{2},\frac{n}{2}+1;\frac{1}{2};\frac{c^2}{c^2-1}\right) + a\left({}_2F_1\left(\frac{1}{2}-\frac{n}{2},\frac{n}{2}+1;\frac{1}{2};\frac{c^2}{c^2-1}\right) - {}_2F_1\left(\frac{1}{2}-\frac{n}{2},\frac{n}{2}+1;-\frac{1}{2};\frac{c^2}{c^2-1}\right)\right)\right), \tag{129}$$

The first five k_{σ_n}s are

$$k_{\sigma_0}(a,b,c) = 1, \tag{130}$$
$$k_{\sigma_1}(a,b,c) = abc, \tag{131}$$
$$k_{\sigma_2}(a,b,c) = a^2b^2\left(2c^2+1\right), \tag{132}$$
$$k_{\sigma_3}(a,b,c) = 3a^3b^3c\left(2c^2+3\right), \tag{133}$$
$$k_{\sigma_4}(a,b,c) = 3a^4b^4(8c^4+24c^2+3), \tag{134}$$
$$k_{\sigma_5}(a,b,c) = 15a^5b^5c(8c^4+40c^2+15). \tag{135}$$

Note that dual activation functions of monomials are also obtained from Theorem 1 by choosing $c_n = 1, c_{n-1} = \cdots = c_0 = 0$. Moreover, obtaining $k_{\sigma_n'}$ is simple either by $\sigma_n(t)' = nt^{n-1}$ or applying Theorem 3 to above expressions on k_{σ_n}.

These are implemented in our code as `Monomial`.

F.8 Gabor

Let us consider a simple version of localized oscillatory activation function given by

$$\sigma_{\text{Gabor}}(t) = \exp(-t^2)\sin(t). \tag{136}$$

The dual actiavtion of Gabor function can be expressed as

$$k_{\text{Gabor}}(a,b,c) = \frac{\exp\left(-\frac{-4a^2b^2c^2+2abc+4ab+a+b}{-8a^2b^2c^2+8ab+4a+4b+2}\right)\left(\exp\left(\frac{2abc}{-4a^2b^2c^2+4ab+2a+2b+1}\right)-1\right)}{\sqrt{-4a^2b^2c^2+a(4b+2)+2b+1}} \tag{137}$$

and that of derivative of Gabor function can be obtained using Theorem 3 as

$$
\begin{aligned}
k_{\mathrm{Gabor}'}(a,b,c) = \exp\left(-\frac{-4a^2b^2c^2+2abc+4ab+a+b}{-8a^2b^2c^2+8ab+4a+4b+2}\right)\times \\
\times\Bigg[\Big(4abc\left(-4a^2b^2c^2+abc+3b+2\right)+2a\left(8ab^2c+6abc+2b+1\right)+2b+1\Big)\times \\
\times\exp\left(\frac{2abc}{-4a^2b^2c^2+4ab+2a+2b+1}\right)+4abc\left(4a^2b^2c^2+abc-3b-2\right)+ \\
+2a\left(-8ab^2c-6abc+2b+1\right)+2b+1\Bigg]\Bigg/\left(-4a^2b^2c^2+a(4b+2)+2b+1\right)^{5/2}. \quad (138)
\end{aligned}
$$

This is implemented in our code as `Gabor`.

F.9 ELU

For Exponential Linear Unit (ELU) [70]

$$\sigma(t) = \mathrm{step}(t)t + \mathrm{step}(-t)(e^t - 1)\,.$$

The $k_\sigma(a,b,c)$ is computed in Tsuchida et al. [48] and we refer to the original paper for the expression.

Note that $k_{\sigma'}$ for ELU has not been computed but Theorem 3 allows to simply obtain it using expression in Tsuchida et al. [48].

G Additional Experiment: Kernel Informed Activation

We explore an activation informed by the normalized Gaussian kernel that achieves the best performance among neural kernels [54]. Although the exact activation is unknown, one can conduct a reverse engineering to find a proper activation whose dual kernel is known and close to the normalized Gaussian. In particular, we focus on the ABReLU activation and recall that its dual kernel is

$$k_{\mathrm{ABReLU}}(a,b,c) = ab\left(\frac{(B-A)^2\left(\sqrt{1-c^2}+\left(\pi-\cos^{-1}(c)\right)c\right)}{2\pi}+ABc\right)$$

for some $A, B \in \mathbb{R}$. Observe that k_{ABReLU} is also homogeneous as like the normalized Gaussian, i.e., $k_\sigma(a,b,c) = ab\cdot\kappa_\sigma(c)$ for $c\in[-1,1]$. We find two slope variables A, B by fitting κ_σ at extreme points, i.e., $\kappa_{\mathrm{ABReLU}}(c) = \exp(c-1)$ for $c=\pm1$. This turns into a quadratic equation and gives us

$$\mathrm{ABReLU}(t) = -0.096\min(t,0) + 1.411\max(t,0)$$

which is illustrated in Figure 2 (**left**). We train a 5-layer ConvNet (known as Myrtle-5 [54]) of 128 width for CIFAR-10 classification. Similar to the CNTK experiment in Section 5, we convert image classes into 10-dimensional one-hot vectors and pre-process CIFAR-10 images with regularized ZCA [54, 58]. We use the SGD optimizer with initial learning rate 0.1, Nesterov momentum with factor 0.9 and ℓ_2 regularizer 0.0005. The batch size is set to 64. The network is trained by minimizing the mean-squared-error (MSE) loss and we report the best test accuracy for 200 epochs. Interestingly, the ABReLU can achieve the highest test accuracy compared to ReLU, GeLU, Erf and parameterized ReLU (PReLU) activations. This supports a connection between infinite width neural kernels and finite width networks in aspect of activation.

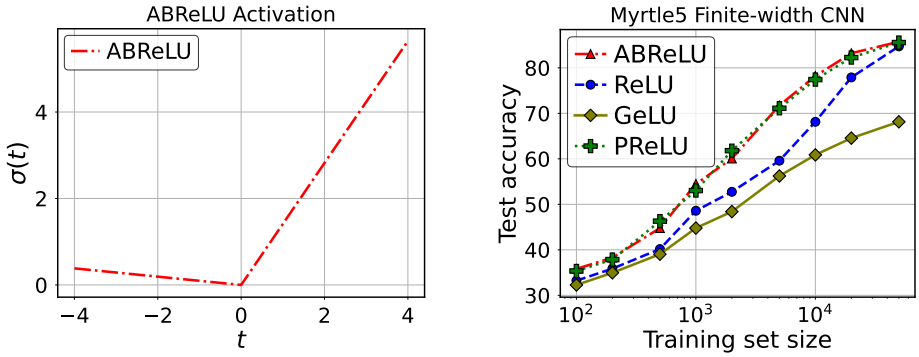

Figure 5: Kernel informed ABReLU (**left**) and test accuracy of finite-width Myrtle5 networks with various activations (**right**).