On the Inductive Bias of Neural Tangent Kernels

Alberto Bietti, Julien Mairal

On the Inductive Bias of Neural Tangent Kernels

Alberto Bietti
Inria*
alberto.bietti@inria.fr

Julien Mairal
Inria*
julien.mairal@inria.fr

Abstract

State-of-the-art neural networks are heavily over-parameterized, making the optimization algorithm a crucial ingredient for learning predictive models with good generalization properties. A recent line of work has shown that in a certain over-parameterized regime, the learning dynamics of gradient descent are governed by a certain kernel obtained at initialization, called the *neural tangent kernel*. We study the inductive bias of learning in such a regime by analyzing this kernel and the corresponding function space (RKHS). In particular, we study smoothness, approximation, and stability properties of functions with finite norm, including stability to image deformations in the case of convolutional networks, and compare to other known kernels for similar architectures.

1 Introduction

The large number of parameters in state-of-the-art deep neural networks makes them very expressive, with the ability to approximate large classes of functions [26, 41]. Since many networks can potentially fit a given dataset, the optimization method, typically a variant of gradient descent, plays a crucial role in selecting a model that generalizes well [39].

A recent line of work [2, 16, 20, 21, 27, 30, 54] has shown that when training deep networks in a certain over-parameterized regime, the dynamics of gradient descent behave like those of a linear model on (non-linear) features determined at initialization. In the over-parameterization limit, these features correspond to a kernel known as the *neural tangent kernel*. In particular, in the case of a regression loss, the obtained model behaves similarly to a minimum norm kernel least squares solution, suggesting that this kernel may play a key role in determining the inductive bias of the learning procedure and its generalization properties. While it is still not clear if this regime is at play in state-of-the-art deep networks, there is some evidence that this phenomenon of "lazy training" [16], where weights only move very slightly during training, may be relevant for early stages of training and for the outmost layers of deep networks [29, 53], motivating a better understanding of its properties.

In this paper, we study the inductive bias of this regime by studying properties of functions in the space associated with the neural tangent kernel for a given architecture (that is, the reproducing kernel Hilbert space, or RKHS). Such kernels can be defined recursively using certain choices of dot-product kernels at each layer that depend on the activation function. For the convolutional case with rectified linear unit (ReLU) activations and arbitrary patches and linear pooling operations, we show that the NTK can be expressed through kernel feature maps defined in a tree-structured hierarchy.

We study smoothness and stability properties of the kernel mapping for two-layer networks and CNNs, which control the variations of functions in the RKHS. In particular, a useful inductive bias when dealing with natural signals such as images is stability of the output to deformations of the input, such as translations or small rotations. A precise notion of stability to deformations was proposed by Mallat [35], and was later studied in [11] in the context of CNN architectures, showing

---

*Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, 38000 Grenoble, France

the benefits of different architectural choices such as small patch sizes. In contrast to the kernels studied in [11], which for instance cover the limiting kernels that arise from training only the last layer of a ReLU CNN, we find that the obtained NTK kernel mappings for the ReLU activation lack a desired Lipschitz property which is needed for stability to deformations in the sense of [11, 12, 35]. Instead, we show that a weaker smoothness property similar to Hölder smoothness holds, and this allows us to show that the kernel mapping is stable to deformations, albeit with a different guarantee.

In order to balance our observations on smoothness, we also consider approximation properties for the NTK of two-layer ReLU networks, by characterizing the RKHS using a Mercer decomposition of the kernel in the basis of spherical harmonics [6, 46, 47]. In particular, we study the decay of eigenvalues for this decomposition, which is then related to the regularity of functions in the space, and provides rates of approximation for Lipschitz functions [6]. We find that the full NTK has better approximation properties compared to other function classes typically defined for ReLU activations [6, 17, 19], which arise for instance when only training the weights in the last layer, or when considering Gaussian process limits of ReLU networks (*e.g.*, [24, 28, 36, 40]).

Contributions. Our main contributions can be summarized as follows:

- We provide a derivation of the NTK for convolutional networks with generic linear operators for patch extraction and pooling, and express the corresponding kernel feature map hierarchically using these operators.
- We study smoothness properties of the kernel mapping for ReLU networks, showing that it is not Lipschitz but satisfies a weaker Hölder smoothness property. For CNNs, we then provide a guarantee on deformation stability.
- We characterize the RKHS of the NTK for two-layer ReLU networks by providing a spectral decomposition of the kernel and studying its spectral decay. This leads to improved approximation properties compared to other function classes based on ReLU.

Related work. Neural tangent kernels were introduced in [27], and similar ideas were used to obtain more quantitative guarantees on the global convergence of gradient descent for over-parameterized neural networks [2, 3, 16, 20, 21, 30, 50, 54]. The papers [3, 20, 51] also derive NTKs for convolutional networks, but focus on simpler architectures. Kernel methods for deep neural networks were studied for instance in [17, 19, 34]. Stability to deformations was originally introduced in the context of the scattering representation [12, 35], and later extended to neural networks through kernel methods in [11]. The inductive bias of optimization in neural network learning was considered, *e.g.*, by [1, 4, 13, 39, 48]. [6, 25, 45, 49] study function spaces corresponding to two-layer ReLU networks. In particular, [25] also analyzes properties of the NTK, but studies a specific high-dimensional limit for generic activations, while we focus on ReLU networks, studying the corresponding eigenvalue decays in finite dimension.

2 Neural Tangent Kernels

In this section, we provide some background on "lazy training" and neural tangent kernels (NTKs), and introduce the kernels that we study in this paper. In particular, we derive the NTK for generic convolutional architectures on ℓ^2 signals. For simplicity of exposition, we consider scalar-valued functions, noting that the kernels may be extended to the vector-valued case, as done, *e.g.*, in [27].

2.1 Lazy training and neural tangent kernels

Multiple recent works studying global convergence of gradient descent in neural networks (*e.g.*, [2, 20, 21, 27, 30, 54]) show that when a network is sufficiently over-parameterized, weights remain close to initialization during training. The model is then well approximated by its linearization around initialization. For a neural network $f(x; \theta)$ with parameters θ and initialization θ_0, we then have:[2]

$$f(x; \theta) \approx f(x; \theta_0) + \langle \theta - \theta_0, \nabla_\theta f(x; \theta_0) \rangle. \tag{1}$$

This regime where weights barely move has also been referred to as "lazy training" [16], in contrast to other situations such as the "mean-field" regime (*e.g.*, [15, 38, 37]), where weights move according

---

[2]While we use gradients in our notations, we note that weak differentiability (*e.g.*, with ReLU activations) is sufficient when studying the limiting NTK [27].

to non-linear dynamics. Yet, with sufficient over-parameterization, the (non-linear) features $x \mapsto \nabla_\theta f(x; \theta_0)$ of the linearized model (1) become expressive enough to be able to perfectly fit the training data, by approximating a kernel method.

Neural Tangent Kernel (NTK). When the width of the network tends to infinity, assuming an appropriate initialization on weights, the features of the linearized model tend to a limiting kernel K, called *neural tangent kernel* [27]:

$$\langle \nabla_\theta f(x; \theta_0), \nabla_\theta f(x', \theta_0) \rangle \to K(x, x'). \tag{2}$$

In this limit and under some assumptions, one can show that the weights move very slightly and the kernel remains fixed during training [27], and that gradient descent will then lead to the minimum norm kernel least-squares fit of the training set in the case of the ℓ_2 loss (see [27] and [37, Section H.7]). Similar interpolating solutions have been found to perform well for generalization, both in practice [10] and in theory [8, 31]. When the number of neurons is large but finite, one can often show that the kernel only deviates slightly from the limiting NTK, at initialization and throughout training, thus allowing convergence as long as the initial kernel matrix is non-degenerate [3, 16, 20, 21].

NTK for two-layer ReLU networks. Consider a two layer network of the form $f(x; \theta) = \sqrt{\frac{2}{m}} \sum_{j=1}^m v_j \sigma(w_j^\top x)$, where $\sigma(u) = (u)_+ = \max(0, u)$ is the ReLU activation, $x \in \mathbb{R}^p$, and $\theta = (w_1^\top, \ldots, w_m^\top, v^\top)$ are parameters with values initialized as $\mathcal{N}(0, 1)$. Practitioners often include the factor $\sqrt{2/m}$ in the variance of the initialization of v_j, but we treat it as a scaling factor following [20, 21, 27], noting that this leads to the same predictions. The factor 2 is simply a normalization constant specific to the ReLU activation and commonly used by practitioners, which avoids vanishing or exploding behavior for deep networks. The corresponding NTK is then given by [16, 21]:

$$K(x, x') = 2(x^\top x') \mathbb{E}_{w \sim \mathcal{N}(0,I)}[\mathbb{1}\{w^\top x \geq 0\}\mathbb{1}\{w^\top x' \geq 0\}] + 2 \mathbb{E}_{w \sim \mathcal{N}(0,I)}[(w^\top x)_+ (w^\top x')_+]$$

$$= \|x\| \|x'\| \kappa \left(\frac{\langle x, x' \rangle}{\|x\| \|x'\|} \right), \tag{3}$$

where

$$\kappa(u) := u\kappa_0(u) + \kappa_1(u) \tag{4}$$

$$\kappa_0(u) = \frac{1}{\pi} \left(\pi - \arccos(u) \right), \qquad \kappa_1(u) = \frac{1}{\pi} \left(u \cdot (\pi - \arccos(u)) + \sqrt{1 - u^2} \right). \tag{5}$$

The expressions for κ_0 and κ_1 follow from standard calculations for arc-cosine kernels of degree 0 and 1 (see [17]). Note that in this two-layer case, the non-linear features obtained for finite neurons correspond to a random features kernel [42], which is known to approximate the full kernel relatively well even with a moderate amount of neurons [7, 42, 43]. One can also extend the derivation to other activation functions, which may lead to explicit expressions for the kernel in some cases [19].

NTK for fully-connected deep ReLU networks. We define a fully-connected neural network by $f(x; \theta) = \sqrt{\frac{2}{m_n}} \langle w^{n+1}, a^n \rangle$, with $a^1 = \sigma(W^1 x)$, and

$$a^k = \sigma \left(\sqrt{\frac{2}{m_{k-1}}} W^k a^{k-1} \right), \quad k = 2, \ldots, n,$$

where $W^k \in \mathbb{R}^{m_k \times m_{k-1}}$ and $w^{n+1} \in \mathbb{R}^{m_n}$ are initialized with i.i.d. $\mathcal{N}(0, 1)$ entries, and $\sigma(u) = (u)_+$ is the ReLU activation and is applied element-wise. Following [27], the corresponding NTK is defined recursively by $K(x, x') = K_n(x, x')$ with $K_0(x, x') = \Sigma_0(x, x') = x^\top x'$, and for $k \geq 1$,

$$\Sigma_k(x, x') = 2 \mathbb{E}_{(u,v) \sim \mathcal{N}(0, B_k)}[\sigma(u)\sigma(v)]$$
$$K_k(x, x') = \Sigma_k(x, x') + 2K_{k-1}(x, x') \mathbb{E}_{(u,v) \sim \mathcal{N}(0, B_k)}[\sigma'(u)\sigma'(v)],$$

3

where $B_k = \begin{pmatrix} \Sigma_{k-1}(x,x) & \Sigma_{k-1}(x,x') \\ \Sigma_{k-1}(x,x') & \Sigma_{k-1}(x',x') \end{pmatrix}$. Using a change of variables and definitions of arc-cosine kernels of degrees 0 and 1 [17], it is easy to show that

$$2\,\mathbb{E}_{(u,v)\sim\mathcal{N}(0,B_k)}[\sigma(u)\sigma(v)] = \sqrt{\Sigma_{k-1}(x,x)\Sigma_{k-1}(x',x')}\,\kappa_1\left(\frac{\Sigma_{k-1}(x,x')}{\sqrt{\Sigma_{k-1}(x,x)\Sigma_{k-1}(x',x')}}\right) \quad (6)$$

$$2\,\mathbb{E}_{(u,v)\sim\mathcal{N}(0,B_k)}[\sigma'(u)\sigma'(v)] = \kappa_0\left(\frac{\Sigma_{k-1}(x,x')}{\sqrt{\Sigma_{k-1}(x,x)\Sigma_{k-1}(x',x')}}\right), \quad (7)$$

where κ_0 and κ_1 are defined in (5).

Feature maps construction. We now provide a reformulation of the previous kernel in terms of explicit feature maps, which provides a representation of the data and makes our study of stability in Section 4 more convenient. For a given input Hilbert space $\mathcal{H}$, we denote by $\varphi_{\mathcal{H},1} : \mathcal{H} \to \mathcal{H}_1$ the kernel mapping into the RKHS $\mathcal{H}_1$ for the kernel $(z,z') \in \mathcal{H}^2 \mapsto \|z\|\|z'\|\kappa_1(\langle z,z'\rangle/\|z\|\|z'\|)$, and by $\varphi_{\mathcal{H},0} : \mathcal{H} \to \mathcal{H}_0$ the kernel mapping into the RKHS $\mathcal{H}_0$ for the kernel $(z,z') \in \mathcal{H}^2 \mapsto \kappa_0(\langle z,z'\rangle/\|z\|\|z'\|)$. We will abuse notation and hide the input space, simply writing φ_1 and φ_0.

Lemma 1 (NTK feature map for fully-connected network). *The NTK for the fully-connected network can be defined as* $K(x,x') = \langle\Phi_n(x),\Phi_n(x')\rangle$, *with* $\Phi_0(x) = \Psi_0(x) = x$ *and for* $k \geq 1$,

$$\Psi_k(x) = \varphi_1(\Psi_{k-1}(x))$$
$$\Phi_k(x) = \begin{pmatrix} \varphi_0(\Psi_{k-1}(x)) \otimes \Phi_{k-1}(x) \\ \varphi_1(\Psi_{k-1}(x)) \end{pmatrix},$$

where $\otimes$ *is the tensor product.*

2.2 Neural tangent kernel for convolutional networks

In this section we study NTKs for convolutional networks (CNNs) on signals, focusing on the ReLU activation. We consider signals in $\ell^2(\mathbb{Z}^d, \mathbb{R}^{m_0})$, that is, signals $x[u]$ with $u \in \mathbb{Z}^d$ denoting the location, $x[u] \in \mathbb{R}^{m_0}$, and $\sum_{u\in\mathbb{Z}^d}\|x[u]\|^2 < \infty$ (for instance, $d = 2$ and $m_0 = 3$ for RGB images). The infinite support allows us to avoid dealing with boundary conditions when considering deformations and pooling. The precise study of ℓ^2 membership is deferred to Section 4.

Patch extraction and pooling operators P^k and A^k. Following [11], we define two linear operators P^k and A^k on $\ell^2(\mathbb{Z}^d)$ for extracting patches and performing (linear) pooling at layer k, respectively. For an $\mathcal{H}$-valued signal $x[u]$, P^k is defined by $P^k x[u] = |S_k|^{-1/2}(x[u+v])_{v\in S_k} \in \mathcal{H}^{|S_k|}$, where S_k is a finite subset of $\mathbb{Z}^d$ defining the patch shape (*e.g.*, a 3x3 box). Pooling is defined as a convolution with a linear filter $h_k[u]$, *e.g.*, a Gaussian filter at scale σ_k as in [11], that is, $A^k x[u] = \sum_{v\in\mathbb{Z}^d} h_k[u-v]x[v]$. In this discrete setting, we can easily include a downsampling operation with factor s_k by changing the definition of A^k to $A^k x[u] = \sum_{v\in\mathbb{Z}^d} h_k[s_k u - v]x[v]$ (in particular, if h_k is a Dirac at 0, we obtain a CNN with "strided convolutions"). In fact, our NTK derivation supports general linear operators $A^k : \ell^2(\mathbb{Z}^d) \to \ell^2(\mathbb{Z}^d)$ on scalar signals.

For defining the NTK feature map, we also introduce the following non-linear point-wise operator M, given for two signals x, y, by

$$M(x,y)[u] = \begin{pmatrix} \varphi_0(x[u]) \otimes y[u] \\ \varphi_1(x[u]) \end{pmatrix}, \quad (8)$$

where $\varphi_{0/1}$ are kernel mappings of arc-cosine 0/1 kernels, as defined in Section 2.1.

CNN definition and NTK. We consider a network $f(x;\theta) = \sqrt{\frac{2}{m_n}}\langle w^{n+1}, a^n\rangle_{\ell^2}$, with

$$\tilde{a}^k[u] = \begin{cases} W^1 P^1 x[u], & \text{if } k = 1, \\ \sqrt{\frac{2}{m_{k-1}}}W^k P^k a^{k-1}[u], & \text{if } k \in \{2,\ldots,n\}, \end{cases}$$
$$a^k[u] = A^k \sigma(\tilde{a}^k)[u], \quad k = 1,\ldots,n,$$

4

where $W^k \in \mathbb{R}^{m_k \times m_{k-1}|S_k|}$ and $w^n \in \ell^2(\mathbb{Z}^d, \mathbb{R}^{m_n})$ are initialized with $\mathcal{N}(0,1)$ entries, and $\sigma(\tilde{x}^k)$ denotes the signal with σ applied element-wise to $\tilde{x}^k$. We are now ready to state our result on the NTK for this model.

Proposition 2 (NTK feature map for CNN). *The NTK for the above CNN, obtained when the number of feature maps $m_1, \ldots, m_n \to \infty$ (sequentially), is given by $K(x, x') = \langle \Phi(x), \Phi(x') \rangle_{\ell^2(\mathbb{Z}^d)}$, with $\Phi(x)[u] = A^n M(x_n, y_n)[u]$, where y_n and x_n are defined recursively for a given input x by $y_1[u] = x_1[u] = P^1 x[u]$, and for $k \geq 2$,*

$$x_k[u] = P^k A^{k-1} \varphi_1(x_{k-1})[u]$$
$$y_k[u] = P^k A^{k-1} M(x_{k-1}, y_{k-1})[u],$$

with the abuse of notation $\varphi_1(x)[u] = \varphi_1(x[u])$ for a signal x.

The proof is given in Appendix A.2, where we also show that in the over-parameterization limit, the pre-activations $\tilde{a}_i^k[u]$ tend to a Gaussian process with covariance $\Sigma^k(x, u; x', u') = \langle x_k[u], x'_k[u'] \rangle$ (this is related to recent papers [24, 40] studying Gaussian process limits of Bayesian convolutional networks). The proof is by induction and relies on similar arguments to [27] for fully-connected networks, in addition to exploiting linearity of the operators P^k and A^k, as well as recursive feature maps for hierarchical kernels. The recent papers [3, 51] also study NTKs for certain convolutional networks; in contrast to these works, our derivation considers general signals in $\ell^2(\mathbb{Z}^d)$, supports intermediate pooling or downsampling by changing A^k, and provides a more intuitive construction through kernel mappings and the operators P^k and A^k. Note that the feature maps x_k are defined independently from the y_k, and in fact correspond to more standard multi-layer deep kernel machines [11, 17, 19, 33] or covariance functions of certain deep Bayesian networks [24, 28, 36, 40]. They can also be seen as the feature maps of the limiting kernel that arises when only training weights in the last layer and fixing other layers at initialization (see, *e.g.*, [19]).

3 Two-Layer Networks

In this section, we study smoothness and approximation properties of the RKHS defined by neural tangent kernels for two-layer networks. For ReLU activations, we show that the NTK kernel mapping is not Lipschitz, but satisfies a weaker smoothness property. In Section 3.2, we characterize the RKHS for ReLU activations and study its approximation properties and benefits. Finally, we comment on the use of other activations in Section 3.3.

3.1 Smoothness of two-layer ReLU networks

Here we study the RKHS $\mathcal{H}$ of the NTK for two-layer ReLU networks, defined in (3), focusing on smoothness properties of the kernel mapping, denoted $\Phi(\cdot)$. Recall that smoothness of the kernel mapping guarantees smoothness of functions $f \in \mathcal{H}$, through the relation

$$|f(x) - f(y)| \leq \|f\|_{\mathcal{H}} \|\Phi(x) - \Phi(y)\|_{\mathcal{H}}. \tag{9}$$

We begin by showing that the kernel mapping for the NTK is not Lipschitz. This is in contrast to the kernel κ_1 in (5), obtained by fixing the weights in the first layer and training only the second layer weights (κ_1 is 1-Lipschitz by [11, Lemma 1]).

Proposition 3 (Non-Lipschitzness). *The kernel mapping $\Phi(\cdot)$ of the two-layer NTK is not Lipschitz:*

$$\sup_{x,y} \frac{\|\Phi(x) - \Phi(y)\|_{\mathcal{H}}}{\|x - y\|} \to +\infty.$$

This is true even when looking only at points x, y on the sphere. It follows that the RKHS $\mathcal{H}$ contains unit-norm functions with arbitrarily large Lipschitz constant.

Note that the instability is due to φ_0, which comes from gradients w.r.t. first layer weigts. We now show that a weaker guarantee holds nevertheless, resembling 1/2-Hölder smoothness.

Proposition 4 (Smoothness for ReLU NTK). *We have the following smoothness properties:*

1. For x, y such that $\|x\| = \|y\| = 1$, the kernel mapping φ_0 satisfies $\|\varphi_0(x) - \varphi_0(y)\| \leq \sqrt{\|x - y\|}$.

2. *For general non-zero x, y, we have $\|\varphi_0(x) - \varphi_0(y)\| \leq \sqrt{\frac{1}{\min(\|x\|, \|y\|)}} \|x - y\|$.*

3. *The kernel mapping Φ of the NTK then satisfies*

$$\|\Phi(x) - \Phi(y)\| \leq \sqrt{\min(\|x\|, \|y\|)\|x - y\|} + 2\|x - y\|.$$

We note that while such smoothness properties apply to the functions in the RKHS of the studied limiting kernels, the neural network functions obtained at finite width and their linearizations around initialization are not in the RKHS and thus may not preserve such smoothness properties, despite preserving good generalization properties, as in random feature models [7, 43]. This discrepancy may be a source of instability to adversarial perturbations.

3.2 Approximation properties for the two-layer ReLU NTK

In the previous section, we found that the NTK κ for two-layer ReLU networks yields weaker smoothness guarantees compared to the kernel κ_1 obtained when the first layer is fixed. We now show that the NTK has better approximation properties, by studying the RKHS through a spectral decomposition of the kernel and the decay of the corresponding eigenvalues. This highlights a tradeoff between smoothness and approximation.

The next proposition gives the Mercer decomposition of the NTK $\kappa(\langle x, u \rangle)$ in (4), where x, y are in the $p - 1$ sphere $\mathbb{S}^{p-1} = \{x \in \mathbb{R}^p : \|x\| = 1\}$. The decomposition is given in the basis of spherical harmonics, as is common for dot-product kernels [46, 47], and our derivation uses results by Bach [6] on similar decompositions of positively homogeneous activations of the form $\sigma_\alpha(u) = (u)_+^\alpha$. See Appendix C for background and proofs.

Proposition 5 (Mercer decomposition of ReLU NTK). *For any $x, y \in \mathbb{S}^{p-1}$, we have the following decomposition of the NTK κ:*

$$\kappa(\langle x, y \rangle) = \sum_{k=0}^{\infty} \mu_k \sum_{j=1}^{N(p,k)} Y_{k,j}(x) Y_{k,j}(y), \tag{10}$$

where $Y_{k,j}, j = 1, \ldots, N(p, k)$ are spherical harmonic polynomials of degree k, and the non-negative eigenvalues μ_k satisfy $\mu_0, \mu_1 > 0$, $\mu_k = 0$ if $k = 2j + 1$ with $j \geq 1$, and otherwise $\mu_k \sim C(p)k^{-p}$ as $k \to \infty$, with $C(p)$ a constant depending only on p. Then, the RKHS is described by:

$$\mathcal{H} = \left\{ f = \sum_{k \geq 0, \mu_k \neq 0} \sum_{j=1}^{N(p,k)} a_{k,j} Y_{k,j}(\cdot) \quad s.t. \quad \|f\|_{\mathcal{H}}^2 := \sum_{k \geq 0, \mu_k \neq 0} \sum_{j=1}^{N(p,k)} \frac{a_{k,j}^2}{\mu_k} < \infty \right\}. \tag{11}$$

The zero eigenvalues prevent certain functions from belonging to the RKHS, namely those with non-zero Fourier coefficients on the corresponding basis elements (note that adding a bias may prevent such zero eigenvalues [9]). Here, a sufficient condition for all such coefficients to be zero is that the function is even [6]. Note that for the arc-cosine 1 kernel κ_1, we have a faster decay $\mu_k = O(k^{-p-2})$, leading to a "smaller" RKHS (see Lemma 17 in Appendix C and [6]). Moreover, the k^{-p} asymptotic equivalent comes from the term $u\kappa_0(u)$ in the definition (4) of κ, which comes from gradients of first layer weights; the second layer gradients yield κ_1, whose contribution to μ_k becomes negligible for large k. We use an identity also used in the recent paper [25] which compares similar kernels in a specific high-dimensional limit for generic activations; in contrast to [25], we focus on ReLUs and study eigenvalue decays in finite dimension. We note that our decomposition uses a uniform distribution on the sphere, which allows a precise study of eigenvalues and approximation properties of the RKHS using spherical harmonics. When the data distribution is also uniform on the sphere, or absolutely continuous w.r.t. the uniform distribution, our obtained eigenvalues are closely related to those of integral operators for learning problems, which can determine, *e.g.*, non-parametric rates of convergence (*e.g.*, [14, 23]) as well as degrees-of-freedom quantities for kernel approximation (*e.g.*, [7, 43]). Such quantities often depend on the eigenvalue decay of the integral operator, which can be obtained from μ_k after taking multiplicity into account. This is also related to the rate of convergence of gradient descent in the lazy training regime, which depends on the minimum eigenvalue of the empirical kernel matrix in [16, 20, 21].

We now provide sufficient conditions for a function $f : \mathbb{S}^{p-1} \to \mathbb{R}$ to be in $\mathcal{H}$, as well as rates of approximation of Lipschitz functions on the sphere, adapting results of [6] (specifically Proposition 2 and 3 in [6]) to our NTK setting.

Corollary 6 (Sufficient condition for $f \in \mathcal{H}$). *Let $f : \mathbb{S}^{p-1} \to \mathbb{R}$ be an even function such that all i-th order derivatives exist and are bounded by η for $0 \leq i \leq s$, with $s \geq p/2$. Then $f \in \mathcal{H}$ with $\|f\|_{\mathcal{H}} \leq C(p)\eta$, where $C(p)$ is a constant that only depends on p.*

Corollary 7 (Approximation of Lipschitz functions). *Let $f : \mathbb{S}^{p-1} \to \mathbb{R}$ be an even function such that $f(x) \leq \eta$ and $|f(x) - f(y)| \leq \eta\|x - y\|$, for all $x, y \in \mathbb{S}^{p-1}$. There is a function $g \in \mathcal{H}$ with $\|g\|_{\mathcal{H}} \leq \delta$, where δ is larger than a constant depending only on p, such that*

$$\sup_{x \in \mathbb{S}^{p-1}} |f(x) - g(x)| \leq C(p)\eta \left(\frac{\delta}{\eta}\right)^{-1/(p/2-1)} \log\left(\frac{\delta}{\eta}\right).$$

For both results, there is an improvement over κ_1, for which Corollary 6 requires $s \geq p/2 + 1$ bounded derivatives, and Corollary 7 leads to a weaker rate in $(\delta/\eta)^{-1/(p/2)}$ (see [6, Propositions 2 and 3] with $\alpha = 1$). These results show that in the over-parameterized regime of the NTK, training multiple layers leads to better approximation properties compared to only training the last layer, which corresponds to using κ_1 instead of κ. In the different regime of "convex neural networks" (*e.g.*, [6, 45]) where neurons can be selected with a sparsity-promoting penalty, the approximation rates shown in [6] for ReLU networks are also weaker than for the NTK in the worst case (though the regime presents benefits in terms of adaptivity), suggesting that perhaps in some situations the "lazy" regime of the NTK could be preferred over the regime where neurons are selected using sparsity.

Homogeneous case. When inputs do not lie on the sphere $\mathbb{S}^{p-1}$ but in $\mathbb{R}^p$, the NTK for two-layer ReLU networks takes the form of a homogeneous dot-product kernel (3), which defines a different RKHS $\bar{\mathcal{H}}$ that we characterize below in terms of the RKHS $\mathcal{H}$ of the NTK on the sphere.

Proposition 8 (RKHS of the homogeneous NTK). *The RKHS $\bar{\mathcal{H}}$ of the kernel $K(x, x') = \|x\|\|x'\|\kappa(\langle x, x'\rangle/\|x\|\|x'\|)$ on $\mathbb{R}^p$ consists of functions of the form $f(x) = \|x\|g(x/\|x\|)$ with $g \in \mathcal{H}$, where $\mathcal{H}$ is the RKHS on the sphere, and we have $\|f\|_{\bar{\mathcal{H}}} = \|g\|_{\mathcal{H}}$.*

Note that while such a restriction to homogeneous functions may be limiting, one may easily obtain non-homogeneous functions by considering an augmented variable $z = (x^\top, R)^\top$ and defining $f(x) = \|z\|g(z/\|z\|)$, where g is now defined on the p-sphere $\mathbb{S}^p$. When inputs are in a ball of radius R, this reformulation preserves regularity properties (see [6, Section 3]).

3.3 Smoothness with other activations

In this section, we look at smoothness of two-layer networks with different activation functions. Following the derivation for the ReLU in Section 2.1, the NTK for a general activation σ is given by

$$K_\sigma(x, x') = \langle x, x'\rangle \, \mathbb{E}_{w \sim \mathcal{N}(0,1)}[\sigma'(\langle w, x\rangle)\sigma'(\langle w, x'\rangle)] + \mathbb{E}_{w \sim \mathcal{N}(0,1)}[\sigma(\langle w, x\rangle)\sigma(\langle w, x'\rangle)].$$

We then have the following the following result.

Proposition 9 (Lipschitzness for smooth activations). *Assume that σ is twice differentiable and that the quantities $\gamma_j := \mathbb{E}_{u \sim \mathcal{N}(0,1)}[(\sigma^{(j)}(u))^2]$ for $j = 0, 1, 2$ are bounded, with $\gamma_0 > 0$. Then, for x, y on the unit sphere, the kernel mapping Φ_σ of K_σ satisfies*

$$\|\Phi_\sigma(x) - \Phi_\sigma(y)\| \leq \sqrt{(\gamma_0 + \gamma_1)\max\left(1, \frac{2\gamma_1 + \gamma_2}{\gamma_0 + \gamma_1}\right)} \cdot \|x - y\|.$$

The proof uses results from [19] on relationships between activations and the corresponding kernels, as well as smoothness results for dot-product kernels in [11] (see Appendix B.3). If, for instance, we consider the exponential activation $\sigma(u) = e^{u-2}$, we have $\gamma_j = 1$ for all j (using results from [19]), so that the kernel mapping is Lipschitz with constant $\sqrt{3}$. For the soft-plus activation $\sigma(u) = \log(1 + e^u)$, we may evaluate the integrals numerically, obtaining $(\gamma_0, \gamma_1, \gamma_2) \approx (2.31, 0.74, 0.11)$, so that the kernel mapping is Lipschitz with constant ≈ 1.75.

4 Deep Convolutional Networks

In this section, we study smoothness and stability properties of the NTK kernel mapping for convolutional networks with ReLU activations. In order to properly define deformations, we consider

continuous signals $x(u)$ in $L^2(\mathbb{R}^d)$ instead of $\ell^2(\mathbb{Z}^d)$ (*i.e.*, we have $\|x\|^2 := \int \|x(u)\|^2 du < \infty$), following [11, 35]. The goal of deformation stability guarantees is to ensure that the data representation (in this case, the kernel mapping Φ) does not change too much when the input signal is slightly deformed, for instance with a small translation or rotation of an image—a useful inductive bias for natural signals. For a C^1-diffeomorphism $\tau : \mathbb{R}^d \to \mathbb{R}^d$, denoting $L_\tau x(u) = x(u - \tau(u))$ the action operator of the diffeomorphism, we will show a guarantee of the form

$$\|\Phi(L_\tau x) - \Phi(x)\| \leq (\omega(\|\nabla\tau\|_\infty) + C\|\tau\|_\infty)\|x\|,$$

where $\|\nabla\tau\|_\infty$ is the maximum operator norm of the Jacobian $\nabla\tau(u)$ over $\mathbb{R}^d$, $\|\tau\|_\infty = \sup_u |\tau(u)|$, ω is an increasing function and C a positive constant. The second term controls translation invariance, and C typically decreases with the scale of the last pooling layer (σ_n below), while the first term controls deformation stability, since $\|\nabla\tau\|_\infty$ measures the "size" of deformations. The function $\omega(t)$ is typically a linear function of t in other settings [11, 35], here we will obtain a faster growth of order $\sqrt{t}$ for small t, due to the weaker smoothness that arises from the arc-cosine 0 kernel mappings.

Properties of the operators. In this continuous setup, P^k is now given for a signal $x \in L^2$ by $P^k x(u) = \lambda(S_k)^{-1/2}(x(u + v))_{v \in S_k}$, where λ is the Lebesgue measure. We then have $\|P^k x\| = \|x\|$, and considering normalized Gaussian pooling filters, we have $\|A^k x\| \leq \|x\|$ by Young's inequality [11]. The non-linear operator M is defined point-wise analogously to (8), and satisfies $\|M(x,y)\|^2 = \|x\|^2 + \|y\|^2$. We thus have that the feature maps in the continuous analog of the NTK construction in Proposition 2 are in L^2 as long as x is in L^2. Note that this does not hold for some smooth activations, where $\|M(x,y)(u)\|$ may be a positive constant even when $x(u) = y(u) = 0$, leading to unbounded L^2 norm for $M(x,y)$. The next lemma studies the smoothness of M, extending results from Section 3.1 to signals in L^2.

Lemma 10 (Smoothness of operator M). *For two signals $x, y \in L^2(\mathbb{R}^d)$, we have*

$$\|M(x,y) - M(x',y')\| \leq \sqrt{\min(\|y\|, \|y'\|)\|x - x'\|} + \|x - x'\| + \|y - y'\|. \tag{12}$$

Assumptions on architecture. Following [11], we introduce an initial pooling layer A^0, corresponding to an anti-aliasing filter, which is necessary for stability and is a reasonable assumption given that in practice input signals are discrete, with high frequencies typically filtered by an acquisition device. Thus, we consider the kernel representation $\Phi_n(x) := \Phi(A^0 x)$, with Φ as in Proposition 2. We also assume that patch sizes are controlled by the scale of pooling filters, that is

$$\sup_{v \in S_k} |v| \leq \beta\sigma_{k-1}, \tag{13}$$

for some constant β, where σ_{k-1} is the scale of the pooling operation A^{k-1}, which typically increases exponentially with depth, corresponding to a fixed downsampling factor at each layer in the discrete case. By a simple induction, we can show the following.

Lemma 11 (Norm and smoothness of Φ_n). *We have $\|\Phi_n(x)\| \leq \sqrt{n+1}\|x\|$, and*

$$\|\Phi_n(x) - \Phi_n(x')\| \leq (n+1)\|x - x'\| + O(n^{5/4})\sqrt{\|x\|\|x - x'\|}.$$

Deformation stability bound. We now present our main guarantee on deformation stability for the NTK kernel mapping (the proof is given in Appendix B).

Proposition 12 (Stability of NTK). *Let $\Phi_n(x) = \Phi(A^0 x)$, and assume $\|\nabla\tau\|_\infty \leq 1/2$. We have the following stability bound:*

$$\|\Phi_n(L_\tau x) - \Phi_n(x)\| \leq \left(C(\beta)^{1/2}Cn^{7/4}\|\nabla\tau\|_\infty^{1/2} + C(\beta)C'n^2\|\nabla\tau\|_\infty + \sqrt{n+1}\frac{C''}{\sigma_n}\|\tau\|_\infty\right)\|x\|,$$

where C, C', C'' are constants depending only on d, and $C(\beta)$ also depends on β defined in (13).

Compared to the bound in [11], the first term shows weaker stability due to faster growth with $\|\nabla\tau\|_\infty$, which comes from (12). The dependence on the depth n is also poorer (n^2 instead of n), however note that in contrast to [11], the norm and smoothness constants of $\Phi_n(x)$ in Lemma 11 grow with n here, partially explaining this gap. We also note that choosing small β (*i.e.*, small patches in a discrete setting) is more helpful to improve stability than a small number of layers n, given that $C(\beta)$ increases polynomially with β, while n typically decreases logarithmically with β when one seeks a fixed target level of translation invariance (see [11, Section 3.2]).
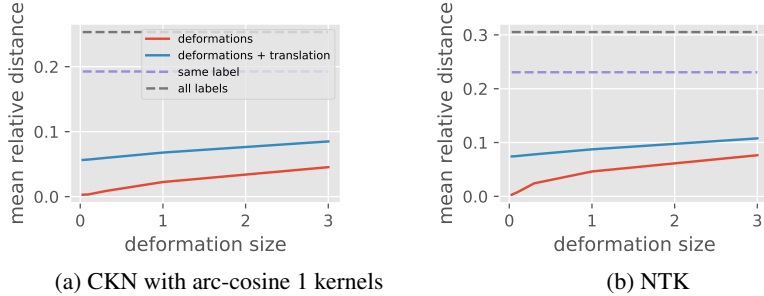
8

Figure 1: Geometry of kernel mapping for CKN and NTK convolutional kernels, on digit images and their deformations from the Infinite MNIST dataset [32]. The curves show average relative distances of a single digit to its deformations, combinations of translations and deformations, digits of the same label, and digits of any label. See Appendix D for more details on the experimental setup.

By fixing weights of all layers but the last, we would instead obtain feature maps of the form $A^n x_n$ (using notation from Proposition 2), which satisfy the improved stability guarantee of [11]. The question of approximation for the deep convolutional case is more involved and left for future work, but it is reasonable to expect that the RKHS for the NTK is at least as large as that of the simpler kernel with fixed layers before the last, given that the latter appears as one of the terms in the NTK. This again hints at a tradeoff between stability and approximation, suggesting that one may be able to learn less stable but more discriminative functions in the NTK regime by training all layers.

Numerical experiments. We now study numerically the stability of (exact) kernel mapping representations for convolutional networks with 2 hidden convolutional layers. We consider both a convolutional kernel network (CKN, [11]) with arc-cosine kernels of degree 1 on patches (corresponding to the kernel obtained when only training the last layer and keeping previous layers fixed) and the corresponding NTK. Figure 1 shows the resulting average distances, when considering collections of digits and deformations thereof. In particular, we find that for small deformations, the distance to the original image tends to grow more quickly for the NTK compared to the CKN, as the theory suggests (a square-root growth rate rather than a linear one). Note also that the relative distances are generally larger for the NTK than for the CKN, suggesting the CKN may be more smooth.

5 Discussion

In this paper, we have studied the inductive bias of the "lazy training" regime for over-parameterized neural networks, by considering the neural tangent kernel of different architectures, and analyzing properties of the corresponding RKHS, which characterizes the functions that can be learned efficiently in this regime. We find that the NTK for ReLU networks has better approximation properties compared to other neural network kernels, but weaker smoothness properties, although these can still guarantee a form of stability to deformations for CNN architectures, providing an important inductive bias for natural signals. While these properties may help obtain better performance when large amounts of data are available, they can also lead to a poorer estimation error when data is scarce, a setting in which smoother kernels or better regularization strategies may be helpful.

It should be noted that while our study of functions in the RKHS may determine what target functions can be learned by over-parameterized networks, the obtained networks with finite neurons do not belong to the same RKHS, and hence may be less stable than such target functions, at least outside of the training data, due to approximations both in the linearization (1) and between the finite neuron and limiting kernels. Additionally, approximation of certain non-smooth functions in this regime may require a very large number of neurons [52]. Finally, we note that while this "lazy" regime is interesting and could partly explain the success of deep learning methods, it does not explain, for instance, the common behavior in early layers where neurons move to select useful features in the data, such as Gabor filters, as pointed out in [16]. In particular, such a behavior might provide better statistical efficiency by adapting to simple structures in the data (see, *e.g.*, [6]), something which is not captured in a kernel regime like the NTK. It would be interesting to study inductive biases in a regime somewhere in between, where neurons may move at least in the first few layers.

Acknowledgments

References

[1] Z. Allen-Zhu, Y. Li, and Y. Liang. Learning and generalization in overparameterized neural networks, going beyond two layers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[2] Z. Allen-Zhu, Y. Li, and Z. Song. A convergence theory for deep learning via over-parameterization. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2019.

[3] S. Arora, S. S. Du, W. Hu, Z. Li, R. Salakhutdinov, and R. Wang. On exact computation with an infinitely wide neural net. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[4] S. Arora, S. S. Du, W. Hu, Z. Li, and R. Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2019.

[5] K. Atkinson and W. Han. *Spherical harmonics and approximations on the unit sphere: an introduction*, volume 2044. Springer Science & Business Media, 2012.

[6] F. Bach. Breaking the curse of dimensionality with convex neural networks. *Journal of Machine Learning Research (JMLR)*, 18(19):1–53, 2017.

[7] F. Bach. On the equivalence between kernel quadrature rules and random feature expansions. *Journal of Machine Learning Research (JMLR)*, 18(21):1–38, 2017.

[8] P. L. Bartlett, P. M. Long, G. Lugosi, and A. Tsigler. Benign overfitting in linear regression. *arXiv preprint arXiv:1906.11300*, 2019.

[9] R. Basri, D. Jacobs, Y. Kasten, and S. Kritchman. The convergence rate of neural networks for learned functions of different frequencies. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[10] M. Belkin, S. Ma, and S. Mandal. To understand deep learning we need to understand kernel learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2018.

[11] A. Bietti and J. Mairal. Group invariance, stability to deformations, and complexity of deep convolutional representations. *Journal of Machine Learning Research (JMLR)*, 20(25):1–49, 2019.

[12] J. Bruna and S. Mallat. Invariant scattering convolution networks. *IEEE Transactions on pattern analysis and machine intelligence (PAMI)*, 35(8):1872–1886, 2013.

[13] Y. Cao and Q. Gu. Generalization bounds of stochastic gradient descent for wide and deep neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[14] A. Caponnetto and E. De Vito. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7(3):331–368, 2007.

[15] L. Chizat and F. Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[16] L. Chizat, E. Oyallon, and F. Bach. On lazy training in differentiable programming. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[17] Y. Cho and L. K. Saul. Kernel methods for deep learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2009.

[18] F. Cucker and S. Smale. On the mathematical foundations of learning. *Bulletin of the American mathematical society*, 39(1):1–49, 2002.

[19] A. Daniely, R. Frostig, and Y. Singer. Toward deeper understanding of neural networks: The power of initialization and a dual view on expressivity. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[20] S. S. Du, J. D. Lee, H. Li, L. Wang, and X. Zhai. Gradient descent finds global minima of deep neural networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2019.

[21] S. S. Du, X. Zhai, B. Poczos, and A. Singh. Gradient descent provably optimizes over-parameterized neural networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[22] C. Efthimiou and C. Frye. *Spherical harmonics in p dimensions*. World Scientific, 2014.

[23] S. Fischer and I. Steinwart. Sobolev norm learning rates for regularized least-squares algorithm. *arXiv preprint arXiv:1702.07254*, 2017.

[24] A. Garriga-Alonso, L. Aitchison, and C. E. Rasmussen. Deep convolutional networks as shallow gaussian processes. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[25] B. Ghorbani, S. Mei, T. Misiakiewicz, and A. Montanari. Linearized two-layers neural networks in high dimension. *arXiv preprint arXiv:1904.12191*, 2019.

[26] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[27] A. Jacot, F. Gabriel, and C. Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[28] J. Lee, Y. Bahri, R. Novak, S. S. Schoenholz, J. Pennington, and J. Sohl-Dickstein. Deep neural networks as gaussian processes. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[29] J. Lee, L. Xiao, S. S. Schoenholz, Y. Bahri, J. Sohl-Dickstein, and J. Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[30] Y. Li and Y. Liang. Learning overparameterized neural networks via stochastic gradient descent on structured data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[31] T. Liang and A. Rakhlin. Just interpolate: Kernel" ridgeless" regression can generalize. *Annals of Statistics*, 2019.

[32] G. Loosli, S. Canu, and L. Bottou. Training invariant support vector machines using selective sampling. In *Large Scale Kernel Machines*, pages 301–320. MIT Press, Cambridge, MA., 2007.

[33] J. Mairal. End-to-End Kernel Learning with Supervised Convolutional Kernel Networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[34] J. Mairal, P. Koniusz, Z. Harchaoui, and C. Schmid. Convolutional kernel networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[35] S. Mallat. Group invariant scattering. *Communications on Pure and Applied Mathematics*, 65(10):1331–1398, 2012.

[36] A. Matthews, M. Rowland, J. Hron, R. E. Turner, and Z. Ghahramani. Gaussian process behaviour in wide deep neural networks. *arXiv preprint arXiv:1804.11271*, 2018.

[37] S. Mei, T. Misiakiewicz, and A. Montanari. Mean-field theory of two-layers neural networks: dimension-free bounds and kernel limit. In *Conference on Learning Theory (COLT)*, 2019.

[38] S. Mei, A. Montanari, and P.-M. Nguyen. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33):E7665–E7671, 2018.

[39] B. Neyshabur, R. Tomioka, and N. Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[40] R. Novak, L. Xiao, Y. Bahri, J. Lee, G. Yang, J. Hron, D. A. Abolafia, J. Pennington, and J. Sohl-Dickstein. Bayesian deep convolutional networks with many channels are gaussian processes. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[41] A. Pinkus. Approximation theory of the mlp model in neural networks. *Acta numerica*, 8:143–195, 1999.

[42] A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems (NIPS)*, 2008.

[43] A. Rudi and L. Rosasco. Generalization properties of learning with random features. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[44] S. Saitoh. *Integral transforms, reproducing kernels and their applications*, volume 369. CRC Press, 1997.

[45] P. Savarese, I. Evron, D. Soudry, and N. Srebro. How do infinite width bounded norm networks look in function space? In *Conference on Learning Theory (COLT)*, 2019.

[46] B. Schölkopf and A. J. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. 2001.

[47] A. J. Smola, Z. L. Ovari, and R. C. Williamson. Regularization with dot-product kernels. In *Advances in Neural Information Processing Systems (NIPS)*, 2001.

[48] D. Soudry, E. Hoffer, M. S. Nacson, S. Gunasekar, and N. Srebro. The implicit bias of gradient descent on separable data. *Journal of Machine Learning Research (JMLR)*.

[49] F. Williams, M. Trager, C. Silva, D. Panozzo, D. Zorin, and J. Bruna. Gradient dynamics of shallow low-dimensional relu networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[50] B. Xie, Y. Liang, and L. Song. Diverse neural network learns true target functions. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.

[51] G. 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.

[52] G. Yehudai and O. Shamir. On the power and limitations of random features for understanding neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[53] C. Zhang, S. Bengio, and Y. Singer. Are all layers created equal? *arXiv preprint arXiv:1902.01996*, 2019.

[54] D. Zou, Y. Cao, D. Zhou, and Q. Gu. Stochastic gradient descent optimizes over-parameterized deep relu networks. *Machine Learning*, 2019.

A   Proofs of NTK derivations

A.1   Proof of Lemma 1

Proof of Lemma 1. By induction, using (6) and (7) and the corresponding definitions of φ_1, φ_0, we can write

$$2\,\mathbb{E}_{(u,v)\sim\mathcal{N}(0,B_k)}[\sigma(u)\sigma(v)] = \langle\varphi_1(\Psi_{k-1}(x)), \varphi_1(\Psi_{k-1}(x'))\rangle$$
$$2\,\mathbb{E}_{(u,v)\sim\mathcal{N}(0,B_k)}[\sigma'(u)\sigma'(v)] = \langle\varphi_0(\Psi_{k-1}(x)), \varphi_0(\Psi_{k-1}(x'))\rangle.$$

The result follows by using the following relation, given three pairs of vectors (x, x'), (y, y') and (z, z') in arbitrary Hilbert spaces:

$$\langle x, x'\rangle + \langle y, y'\rangle\langle z, z'\rangle = \langle\begin{pmatrix} y\otimes z \\ x \end{pmatrix}, \begin{pmatrix} y'\otimes z' \\ x' \end{pmatrix}\rangle$$

$\square$

A.2   Proof of Proposition 2 (NTK for CNNs)

In this section, we will denote by x_k, y_k (resp x_k', y_k') the feature maps associated to an input x (resp x'), as defined in Proposition 2. We follow the proofs of Jacot et al. [27, Proposition 1 and Theorem 1].

We begin by proving the following lemma, which characterizes the Gaussian process behavior of the pre-activations $\tilde{a}_i^k[u]$, seen as a function of x and u, in the over-parameterization limit.

Lemma 13. *As $m_1, \ldots, m_{n-1} \to \infty$, the pre-activations $\tilde{a}_i^k[u]$ for $k = 1, \ldots, n$ tend (in law) to i.i.d. centered Gaussian processes with covariance*

$$\Sigma^k(x, u; x', u') = \langle x_k[u], x_k'[u']\rangle. \tag{14}$$

Proof. We show this by induction. For $k = 1$, $\tilde{a}_i^1[u]$ is clearly Gaussian, and we have

$$\Sigma^1(x, u; x', u') = \mathbb{E}[\tilde{a}_i^1[u]\tilde{a}_i'^1[u']]$$
$$= \mathbb{E}[(W^1 P^1 x[u])_i(W^1 P^1 x'[u'])_i].$$

Writing $W_{ij}^k \in \mathbb{R}^{|S_k|}$ the vector of weights for the filter associated to the input feature map j and output feature map i, we have $(W^1 P^1 x[u])_i = \sum_{j=1}^{m_1} W_{ij}^{1\top} P^1 x_j[u]$. Then we have

$$\Sigma^k(x, u; x', u') = \sum_{j,j'} \mathbb{E}[W_{ij}^{1\top} P^1 x_j[u] P^1 x_{j'}'[u']^\top W_{ij'}^1]$$
$$= \sum_{j,j'} \text{Tr}(\mathbb{E}[W_{ij'}^1, W_{ij}^{1\top}] P^1 x_j[u] P^1 x_{j'}'[u']^\top)$$
$$= \sum_j \text{Tr}(P^1 x_j[u] P^1 x_j'[u']^\top) = \langle P^1 x[u], P^1 x'[u']\rangle = \langle P^1 x_0[u], P^1 x_0'[u']\rangle,$$

by noticing that $\mathbb{E}[W_{ij'}^1, W_{ij}^{1\top}] = \delta_{j,j'} I_{|S_1|}$.

Now, for $k \geq 2$, we have by similar arguments that conditioned on a^{k-1}, $\tilde{a}_i^k[u]$ is Gaussian, with covariance

$$\mathbb{E}[\tilde{a}_i^k[u]\tilde{a}_i'^k[u']|a^{k-1}, a'^{k-1}] = \frac{2}{m_{k-1}} \sum_j \langle P^k a_j^{k-1}[u], P^k a_j'^{k-1}[u']\rangle.$$

By the inductive hypothesis, $\tilde{a}_j^{k-1}[u]$ as a function of x and u tend to Gaussian processes in the limit $m_1, \ldots, m_{k-2} \to \infty$. By the law of large numbers, we have, as $m_{k-1} \to \infty$,

$$\mathbb{E}[\tilde{a}_i^k[u]\tilde{a}_i'^k[u']|a^{k-1}, a'^{k-1}]$$
$$\to \Sigma^k(x, u; x', u') := 2\,\mathbb{E}_{f\sim GP(0,\Sigma^{k-1})}[\langle P^k A^{k-1}\sigma(f(x))[u], P^k A^{k-1}\sigma(f(x'))[u']\rangle].$$

13

Since this covariance is deterministic, the pre-activations $\tilde{a}_i^k[u]$ are also unconditionally a Gaussian process in the limit, with covariance Σ^k.

Now it remains to show that

$$2\,\mathbb{E}_{f\sim GP(0,\Sigma^{k-1})}[\langle P^k A^{k-1}\sigma(f(x))[u], P^k A^{k-1}\sigma(f(x'))[u']\rangle]$$
$$= \langle P^k A^{k-1}\varphi_1(x_{k-1})[u], P^k A^{k-1}\varphi_1(x'_{k-1})[u']\rangle.$$

Notice that by linearity of P^k and A^{k-1}, it suffices to show

$$2\,\mathbb{E}_{f\sim GP(0,\Sigma^{k-1})}[\sigma(f(x))[v]\sigma(f(x'))[v']] = \langle\varphi_1(x_{k-1})[v],\varphi_1(x'_{k-1})[v']\rangle$$
$$= \|x_{k-1}[v]\|\|x'_{k-1}[v']\|\kappa_1\left(\frac{\langle x_{k-1}[v], x_{k-1}[v']\rangle}{\|x_{k-1}[v]\|\|x'_{k-1}[v']\|}\right),$$

for any v, v' (the last equality follows from the definition of φ_1). Noting that $(\sigma(f(x))[v], \sigma(f(x'))[v']) = (\sigma(f(x)[v]), \sigma(f(x')[v']))$ when $f \sim GP(0, \Sigma^{k-1})$ has Gaussian distribution with zero mean and covariance $\begin{pmatrix} \Sigma^{k-1}(x,v;x,v) & \Sigma^{k-1}(x,v;x',v') \\ \Sigma^{k-1}(x',v';x',v') & \Sigma^{k-1}(x',v';x',v') \end{pmatrix}$, the results follow from (6) and (14) for k–1 by the inductive hypothesis. $\square$

We now state and prove a lemma which covers the recursion in the NTK for convolutional layers (*i.e.*, up to the last fully-connected layer).

Lemma 14. *As $m_1, \ldots, m_{n-1} \to \infty$, the gradients of the pre-activations, $\nabla_\theta \tilde{a}_i^k[u]$, for $k = 1, \ldots, n$ satisfy*

$$\langle\nabla_\theta\tilde{a}_i^k[u], \nabla_\theta\tilde{a}_{i'}^{\prime k}[u']\rangle \to \delta_{i,i'}\tilde{\Gamma}_\infty^k(x,u;x',u') = \delta_{i,i'}\langle y_k[u], y'_k[u']\rangle.$$

Proof. We prove this by induction. For $k = 1$, denoting by W_i^1 the ith row of W^1, we have

$$\langle\nabla_\theta\tilde{a}_i^1[u], \nabla_\theta\tilde{a}_{i'}^{\prime 1}[u']\rangle = \langle\nabla_{W^1}(W^1 P^1 x[u])_i, \nabla_{W^1}(W^1 P^1 x'[u'])_{i'}\rangle$$
$$= \sum_s\langle\nabla_{W_s^1}W_i^1 P^1 x[u], \nabla_{W_s^1}W_{i'}^1 P^1 x'[u']\rangle$$
$$= \delta_{i,i'}\langle P^1 x[u], P^1 x'[u']\rangle.$$

For $k \geq 2$, assume the result holds up to k–1. We have

$$\langle\nabla_\theta\tilde{a}_i^k[u], \nabla_\theta\tilde{a}_{i'}^{\prime k}[u']\rangle = \langle\nabla_{W^k}\tilde{a}_i^k[u], \nabla_{W^k}\tilde{a}_{i'}^{\prime k}[u']\rangle + \langle\nabla_{W^{1:k-1}}\tilde{a}_i^k[u], \nabla_{W^{1:k-1}}\tilde{a}_{i'}^{\prime k}[u']\rangle.$$

For the first term, we have, as in the $k = 1$ case,

$$\langle\nabla_{W^k}\tilde{a}_i^k[u], \nabla_{W^k}\tilde{a}_{i'}^{\prime k}[u']\rangle = \frac{2\delta_{i,i'}}{m_{k-1}}\langle P^k a^{k-1}[u], P^k a^{\prime k-1}[u']\rangle$$
$$= \frac{2\delta_{i,i'}}{m_{k-1}}\sum_j\langle P^k a_j^{k-1}[u], P^k a_j^{\prime k-1}[u']\rangle$$
$$= \frac{2\delta_{i,i'}}{m_{k-1}}\sum_j\langle P^k A^{k-1}\sigma(\tilde{a}_j^{k-1})[u], P^k A^{k-1}\sigma(\tilde{a}_j^{\prime k-1})[u']\rangle.$$

When $m_1, \ldots, m_{k-2} \to \infty$, $\tilde{a}_j^{k-1}[u]$ tends to a Gaussian process with covariance Σ^{k-1} by Lemma 13, and when $m_{k-1} \to \infty$, the quantity above converges to its expectation:

$$\langle\nabla_{W^k}\tilde{a}_i^k[u], \nabla_{W^k}\tilde{a}_{i'}^{\prime k}[u']\rangle \to 2\delta_{i,i'}\,\mathbb{E}_{f\sim GP(0,\Sigma^{k-1})}[\langle P^k A^{k-1}\sigma(f(x))[u], P^k A^{k-1}\sigma(f(x))[u']\rangle]$$
$$= \delta_{i,i'}\langle P^k A^{k-1}\varphi_1(x_{k-1})[u], P^k A^{k-1}\varphi_1(x'_{k-1})[u']\rangle,$$

by using similar arguments to the proof of Lemma 13.

For the second term, identifying all parameters $W^{1:k-1}$ with a vector $\hat{\theta} \in \mathbb{R}^q$, we have by linearity and the chain rule:

$$\sqrt{m_{k-1}}\nabla_{\hat{\theta}}\tilde{a}_i^k[u] = \sum_j\nabla_{\hat{\theta}}(W_{ij}^{k\top}P^k A^{k-1}\tilde{a}_j^{k-1}[u])$$
$$= \sum_j P^k A^{k-1}y_j^{k-1}[u]^\top \cdot W_{ij}^k \in \mathbb{R}^q,$$

where $y_j^{k-1}[u] := \sqrt{2}\sigma'(\tilde{a}_j^{k-1}[u])\nabla_{\hat{\theta}}\tilde{a}_j^{k-1}[u] \in \mathbb{R}^q$. Here we have identified $P^k A^{k-1} y_j^{k-1}[u]^\top$ with a matrix in $\mathbb{R}^{q \times |S_k|}$, where columns are given by $|S_k|^{-1/2} A^{k-1} y_j^{k-1}[u+v] \in \mathbb{R}^q$, indexed by $v \in S_k$. We thus have

$$\langle \nabla_{W^{1:k-1}}\tilde{a}_i^k[u], \nabla_{W^{1:k-1}}\tilde{a}_{i'}^{\prime k}[u']\rangle = \frac{1}{m_{k-1}} \sum_{j,j'} W_{i,j}^{k\top} \underbrace{(P^k A^{k-1} y_j^{k-1}[u] \cdot P^k A^{k-1} y_{j'}^{\prime k-1}[u']^\top)}_{=:\Pi^{j,j'} \in \mathbb{R}^{|S_k| \times |S_k|}} W_{i',j'}^k.$$

For $v, v' \in S_k$, when $m_1, \ldots, m_{k-2} \to \infty$ and using the inductive hypothesis, we have

$$\Pi_{v,v'}^{j,j'} = \frac{1}{|S_k|} A^{k-1} y_j^{k-1}[u+v]^\top A^{k-1} y_{j'}^{\prime k-1}[u'+v']$$

$$\to \delta_{j,j'} \bar{\Pi}_{v,v'}^j := \frac{\delta_{j,j'}}{|S_k|} \langle A^{k-1} \gamma_j^{k-1}[u+v], A^{k-1} \gamma_j^{\prime k-1}[u'+v']\rangle,$$

where $\gamma_j^{k-1}[u] := \sqrt{2}\sigma'(\tilde{a}_j^{k-1}[u])y_{k-1}[u]$. Indeed, by linearity it suffices to check that $y_j^{k-1}[u]^\top y_{j'}^{\prime k-1}[u'] = 2\delta_{j,j'}\sigma'(\tilde{a}_j^{k-1}[u])\sigma'(\tilde{a}_{j'}^{\prime k-1}[u'])\langle y_{k-1}[u], y_{k-1}'[u']\rangle$ for any j, j', u, u', which is true by the inductive hypothesis. In this same limit (with m_{k-1} fixed), we then have

$$\langle \nabla_{W^{1:k-1}}\tilde{a}_i^k[u], \nabla_{W^{1:k-1}}\tilde{a}_{i'}^{\prime k}[u']\rangle \to \frac{1}{m_{k-1}} \sum_j W_{i,j}^{k\top} \bar{\Pi}^j W_{i',j}^k.$$

When $m_{k-1} \to \infty$, by the law of large numbers, this quantity converges to its expectation:

$$\langle \nabla_{W^{1:k-1}}\tilde{a}_i^k[u], \nabla_{W^{1:k-1}}\tilde{a}_{i'}^{\prime k}[u']\rangle \to \text{Tr}(\mathbb{E}[W_{i',1}^k W_{i,1}^{k\top}]\Pi^\infty) = \delta_{i,i'} \text{Tr}(\Pi^\infty),$$

where Π^∞ is given by

$$\Pi_{v,v'}^\infty = \frac{1}{|S_k|} \langle A^{k-1} \gamma_\infty^{k-1}[u+v], A^{k-1} \gamma_\infty^{\prime k-1}[u'+v']\rangle,$$

with $\gamma_\infty^{k-1}[u] = \varphi_0(x_{k-1}[u]) \otimes y_{k-1}[u]$. Indeed, using Lemma 13 and linearity of A^{k-1}, it is enough to check that

$$2\,\mathbb{E}_{f \sim GP(0,\Sigma^{k-1})}[\sigma'(f(x)[u])\sigma'(f(x')[u'])\langle y_{k-1}[u], y_{k-1}'[u']\rangle] = \langle \gamma_\infty^{k-1}[u], \gamma_\infty^{\prime k-1}[u']\rangle,$$

which holds by definition of φ_0 and Σ^{k-1}.

Finally, notice that

$$\text{Tr}(\Pi^\infty) = \frac{1}{|S_k|} \sum_{v \in S_k} \langle A^{k-1} \gamma_\infty^{k-1}[u+v], A^{k-1} \gamma_\infty^{\prime k-1}[u'+v]\rangle$$

$$= \langle P^k A^{k-1} \gamma_\infty^{k-1}[u], P^k A^{k-1} \gamma_\infty^{\prime k-1}[u']\rangle.$$

Thus we have

$$\tilde{\Gamma}_\infty(x, u, x', u') = \langle P^k A^{k-1} \varphi_1(x_{k-1})[u], P^k A^{k-1} \varphi_1(x_{k-1}')[u']\rangle + \langle P^k A^{k-1} \gamma_\infty^{k-1}[u], P^k A^{k-1} \gamma_\infty^{\prime k-1}[u']\rangle$$

$$= \langle P^k A^{k-1} M(x_{k-1}, y_{k-1})[u], P^k A^{k-1} M(x_{k-1}', y_{k-1}')[u']\rangle$$

$$= \langle y_k[u], y_k'[u']\rangle,$$

which concludes the proof.

$\square$

Armed with the two above lemmas, we can now prove Proposition 2 by studying the gradient of the prediction layer.

Proof of Proposition 2. We have

$$\langle \nabla_\theta f(x; \theta), \nabla_\theta f(x'; \theta)\rangle = \langle \nabla_{w^{n+1}} f(x; \theta), \nabla_{w^{n+1}} f(x'; \theta)\rangle + \langle \nabla_{W^{1:n}} f(x; \theta), \nabla_{W^{1:n}} f(x'; \theta)\rangle$$

The first term writes

$$\langle \nabla_{w^{n+1}} f(x;\theta), \nabla_{w^{n+1}} f(x';\theta) \rangle = \frac{2}{m_n} \sum_j \langle a_j^n, a_j'^n \rangle$$

$$= \frac{2}{m_n} \sum_j \sum_u \langle A^n \sigma(\tilde{a}_j^n)[u], A^n \sigma(\tilde{a}_j'^n)[u] \rangle$$

Using similar arguments as in the above proofs and using Lemma 13, as $m_1, \dots, m_n \to \infty$, we have

$$\langle \nabla_{w^{n+1}} f(x;\theta), \nabla_{w^{n+1}} f(x';\theta) \rangle \to \sum_u \langle A^n \varphi_1(x_n)[u], A^n \varphi_1(x'_n)[u] \rangle = \langle A^n \varphi_1(x_n), A^n \varphi_1(x'_n) \rangle.$$

For the second term, we have

$$\langle \nabla_{W^{1:n}} f(x;\theta), \nabla_{W^{1:n}} f(x';\theta) \rangle = \frac{2}{m_n} \sum_{u,u'} \sum_{j,j'} w_j^{n+1}[u] w_{j'}^{n+1}[u'] \langle \nabla_{W^{1:n}} a_j^n[u], \nabla_{W^{1:n}} a_{j'}'^n[u'] \rangle.$$

We can use similar arguments to the proof of Lemma 14 to show that when $m_1, \dots, m_n \to \infty$, we have

$$\langle \nabla_{W^{1:n}} f(x;\theta), \nabla_{W^{1:n}} f(x';\theta) \rangle \to \sum_{u,u'} \mathbb{E}[w_1^{n+1}[u] w_1^{n+1}[u']] \langle A^n \gamma_n[u], A^n \gamma'_n[u'] \rangle = \langle A^n \gamma_n, A^n \gamma'_n \rangle,$$

where $\gamma_n[u] := \varphi_0(x_n[u]) \otimes y_n[u]$.

The final result follows by combining both terms. $\qquad\square$

B Proofs for Smoothness and Stability to Deformations

B.1 Proof of Proposition 3

Proof. Using notations from Section 2.1, we can write

$$\kappa(u) = u\kappa_0(u) + \kappa_1(u) = \frac{u}{\pi}(\pi - \arccos(u)) + \frac{1}{\pi}(u(\pi - \arccos(u)) + \sqrt{1 - u^2}).$$

For $\|x\| = \|y\| = 1$, and denoting $u = \langle x, y \rangle$, we have

$$\frac{\|\Phi(x) - \Phi(y)\|^2}{\|x - y\|^2} = \frac{2\kappa(1) - 2\kappa(u)}{2 - 2u}$$

$$= \frac{\kappa_0(1) - u\kappa_0(u)}{1 - u} + \frac{\kappa_1(1) - \kappa_1(u)}{1 - u}$$

$$\sim_{u \to 1^-} u\kappa_0'(u) + \kappa_0(u) + \kappa_1'(u) \xrightarrow{u \to 1^-} +\infty,$$

where the equivalent follows from l'Hôpital's rule, and we have $\kappa_0'(u) = 1/\pi\sqrt{1 - u^2} \to +\infty$, while $\kappa_0(1) = \kappa_1(1) = \kappa_1'(1) = 1$. It follows that the supremum over x, y is unbounded.

For the second part, fix an arbitrary $L > 0$. We can find x, y such that $\|\Phi(x) - \Phi(y)\|_{\mathcal{H}} > L\|x - y\|$. Take

$$f = \frac{\Phi(x) - \Phi(y)}{\|\Phi(x) - \Phi(y)\|_{\mathcal{H}}} \in \mathcal{H}.$$

We have $\|f\|_{\mathcal{H}} = 1$ and $f(x) - f(y) = \|\Phi(x) - \Phi(y)\|_{\mathcal{H}} > L\|x - y\|$, so that the Lipschitz constant of f is larger than L. $\qquad\square$

B.2 Proof of Proposition 4 (smoothness of 2-layer ReLU NTK)

Proof. Denoting $u = \langle x, y \rangle$, we have

$$\frac{\|\varphi_0(x) - \varphi_0(y)\|^2}{\|x - y\|} = \frac{2\kappa_0(1) - 2\kappa_0(u)}{\sqrt{2 - 2u}}.$$

As a function of $u \in [-1, 1]$, this quantity decreases from 1 to $1/2\pi$, and is thus upper bounded by 1, proving the first part.

Note that if u, v are on the sphere and $\alpha \geq 1$, then $\|u - \alpha v\| \geq \|u - v\|$. This yields

$$\|x - y\| \geq \min(\|x\|, \|y\|)\|\bar{x} - \bar{y}\|,$$

where $\bar{x}, \bar{y}$ denote the normalized vectors. Then, noting that φ_0 is 0-homogeneous, we have

$$\frac{\|\varphi_0(x) - \varphi_0(y)\|^2}{\|x - y\|} = \frac{\|\varphi_0(\bar{x}) - \varphi_0(\bar{y})\|^2}{\|x - y\|}$$
$$\leq \frac{\|\varphi_0(\bar{x}) - \varphi_0(\bar{y})\|^2}{\min(\|x\|, \|y\|)\|\bar{x} - \bar{y}\|}$$
$$\leq \frac{1}{\min(\|x\|, \|y\|)},$$

by using the previous result on the sphere, and the result for the second part follows.

For the last part, assume x has smaller norm than y. We have

$$\|\Phi(x) - \Phi(y)\| = \left\| \begin{pmatrix} \varphi_0(x) \otimes x \\ \varphi_1(x) \end{pmatrix} - \begin{pmatrix} \varphi_0(y) \otimes y \\ \varphi_1(y) \end{pmatrix} \right\|$$
$$\leq \left\| \begin{pmatrix} \varphi_0(x) \otimes x \\ \varphi_1(x) \end{pmatrix} - \begin{pmatrix} \varphi_0(y) \otimes x \\ \varphi_1(y) \end{pmatrix} \right\| + \left\| \begin{pmatrix} \varphi_0(y) \otimes x \\ \varphi_1(y) \end{pmatrix} - \begin{pmatrix} \varphi_0(y) \otimes y \\ \varphi_1(y) \end{pmatrix} \right\|$$
$$= \sqrt{\|x\|^2\|\varphi_0(x) - \varphi_0(y)\|^2 + \|\varphi_1(x) - \varphi_1(y)\|^2} + \|\varphi_0(y)\|\|x - y\|$$
$$\leq \sqrt{\|x\|\|x - y\| + \|x - y\|^2} + \|x - y\| \leq \sqrt{\|x\|\|x - y\|} + 2\|x - y\|,$$

where in the last line we used $\|\varphi_0(y)\| = 1$, $\|\varphi_0(x) - \varphi_0(y)\|^2 \leq \|x - y\|/\|x\|$, as well as $\|\varphi_1(x) - \varphi_1(y)\| \leq \|x - y\|$, which follows from [11, Lemma 1]. We conclude by symmetry. $\square$

B.3 Proof of Proposition 9 (smooth activations)

Proof. We introduce the following kernels defined on the sphere:

$$\kappa_j(\langle x, x' \rangle) = \mathbb{E}_{w \sim \mathcal{N}(0, I)}[\sigma^{(j)}(\langle w, x \rangle)\sigma^{(j)}(\langle w, x' \rangle)].$$

Note that these are indeed dot-product kernels, defined as polynomial expansions in terms of the squared Hermite expansion coefficients of $\sigma^{(j)}$, as shown by Daniely [19] (called "dual activations"). In fact, [19, Lemma 11] also shows that the mapping from activation to dual activation commutes with differentiation, so that $\kappa_j(u) = \kappa_0^{(j)}(u)$, for $u \in (-1, 1)$. The assumption made in this proposition implies that the j-th order derivatives of κ_0 as $u \to 1$ exist, with $\kappa_0^{(j)}(1) = \kappa_j(1) = \gamma_j < +\infty$.

Then, the NTK on the sphere takes the form $K_\sigma(x, x') = \kappa_\sigma(\langle x, x' \rangle)$, where $\kappa_\sigma(u) = u\kappa_1(u) + \kappa_0(u)$. Then, if we consider the kernel $\hat{\kappa}_\sigma(u) = \frac{\kappa_\sigma(u)}{\kappa_\sigma(1)} = \frac{\kappa_\sigma(u)}{\gamma_0 + \gamma_1}$, we have

$$\hat{\kappa}_\sigma(1) = 1 \quad \text{and} \quad \hat{\kappa}'_\sigma(1) = \frac{\kappa_1(1) + \kappa_1'(1) + \kappa_0'(1)}{\gamma_0 + \gamma_1} = \frac{\gamma_2 + 2\gamma_1}{\gamma_0 + \gamma_1}.$$

Applying Lemma 1 of [11] to this kernel, and re-multiplying by $\gamma_0 + \gamma_1$ yields the final result. $\square$

B.4 Proof of Lemma 10 (smoothness of operator M in $L^2(\mathbb{R}^d)$)

Proof. Using similar arguments as in the proof of Proposition 4, we can show that for any $u \in \mathbb{R}^d$

$$\|M(x, y)(u) - M(x', y')(u)\| \leq \sqrt{\min(\|y(u)\|, \|y'(u)\|)\|x(u) - x'(u)\|}$$
$$+ \|x(u) - x'(u)\| + \|y(u) - y'(u)\|$$

17

Now assume that $\min(\|y\|, \|y'\|) = \|y\|$. By the triangle inequality in $L^2(\mathbb{R}^d)$, we then have

$$\|M(x,y) - M(x',y')\| \le \sqrt{\int \min(\|y(u)\|, \|y'(u)\|)\|x(u) - x'(u)\|du} + \|x - x'\| + \|y - y'\|$$

$$\le \sqrt{\int \|y(u)\|\|x(u) - x'(u)\|du} + \|x - x'\| + \|y - y'\|$$

$$\le \sqrt{\|y\|\|x - x'\|} + \|x - x'\| + \|y - y'\|,$$

where the last inequality follows from Cauchy-Schwarz. We obtain the final result by symmetry. $\square$

B.5 Proof of Proposition 12 (stability to deformations)

We first recall the following results from [11].

Lemma 15 ([11]). *Assume $\|\nabla\tau\|_\infty \le 1/2$, and $\sup_{v \in S_k} |v| \le \beta\sigma_{k-1}$ for all k. We have*

$$\|[P^k A^{k-1}, L_\tau]\| \le C(\beta)\|\nabla\tau\|_\infty$$

$$\|L^\tau A^n - A^n\| \le \frac{C_2}{\sigma_n}\|\tau\|_\infty,$$

where $C(\beta)$ grows with β as β^{d+1}.

We are now ready to prove Proposition 12.

Proof. In order to compare $\Phi_n(L_\tau x)$ and $\Phi_n(x)$, we introduce intermediate sequences of feature maps, denoted $x_k^{(k_0)}$ and $y_k^{(k_0)}$, where the deformation operator L_τ acts at layer k_0. In particular, we denote by $x_k^{(0)}$, $y_k^{(0)}$ the feature maps obtained for the input $L_\tau x$, and if $k_0 \ge 1$, we define $x_k^{(k_0)} = x_k$, $y_k^{(k_0)} = y_k$ for $k \le k_0$,

$$x_{k_0+1}^{(k_0)} = P^{k_0+1} A^{k_0} L_\tau \varphi_1(x_{k_0})$$

$$y_{k_0+1}^{(k_0)} = P^{k_0+1} A^{k_0} L_\tau M(x_{k_0}, y_{k_0}),$$

and for $k \ge k_0 + 2$,

$$x_k^{(k_0)} = P^k A^{k-1} L_\tau \varphi_1(x_{k-1}^{k_0})$$

$$y_k^{(k_0)} = P^k A^{k-1} L_\tau M(x_{k-1}^{k_0}, y_{k-1}^{k_0}).$$

Then, we have the following

$$\|\Phi_n(L_\tau x) - \Phi_n(x)\| = \|A^n M(x_n^{(0)}, y_n^{(0)}) - A^n M(x_n, y_n)\|$$

$$\le \|A^n M(x_n^{(0)}, y_n^{(0)}) - A^n L_\tau M(x_n, y_n)\|$$

$$+ \|A^n L_\tau M(x_n, y_n) - A^n M(x_n, y_n)\|$$

$$\le \|A^n M(x_n^{(0)}, y_n^{(0)}) - A^n L_\tau M(x_n, y_n)\|$$

$$+ \|A^n L_\tau - A^n\|\|M(x_n, y_n)\|.$$

Using Lemma 15, we have

$$\|A^n L_\tau - A^n\| \le \|[A^n, L_\tau]\| + \|L_\tau A^n - A^n\|$$

$$\le C(\beta)\|\nabla\tau\|_\infty + \frac{C_2}{\sigma_n}\|\tau\|_\infty.$$

Separately, we have $\|M(x_n, y_n)\|^2 = \|x_n\|^2 + \|y_n\|^2 \le (n+1)\|x\|^2$, so that

$$\|\Phi_n(L_\tau x) - \Phi_n(x)\| \le \|A^n M(x_n^{(0)}, y_n^{(0)}) - A^n L_\tau M(x_n, y_n)\|$$

$$+ \sqrt{n+1}\left(C(\beta)\|\nabla\tau\|_\infty + \frac{C_2}{\sigma_n}\|\tau\|_\infty\right)\|x\|.$$

18

We now bound the first term above by induction. For $n = 1$, we have

$$\|A^1 M(x_1^{(0)}, y_1^{(0)}) - A^1 L_\tau M(x_1, y_1)\| \leq \|A^1 M(x_1^{(0)}, y_1^{(0)}) - A^1 L_\tau M(x_1, y_1)\|$$
$$\leq \|M(x_1^{(0)}, y_1^{(0)}) - L_\tau M(x_1, y_1)\|$$
$$= \|M(L_\tau x, L_\tau x) - L_\tau M(x, y)\| = 0,$$

by noting that M is a point-wise operator, and thus commutes with L_τ. We now assume $n \geq 2$. We have

$$\|A^n M(x_n^{(0)}, y_n^{(0)}) - A^n L_\tau M(x_n, y_n)\| \leq \|M(x_n^{(0)}, y_n^{(0)}) - L_\tau M(x_n, y_n)\|$$
$$= \|M(x_n^{(0)}, y_n^{(0)}) - M(L_\tau x_n, L_\tau y_n)\|$$
$$\leq \sqrt{\|L_\tau y_n\| \|x_n^{(0)} - L_\tau x_n\|}$$
$$+ \|x_n^{(0)} - L_\tau x_n\| + \|y_n^{(0)} - L_\tau y_n\|,$$

where we used Lemma 10 and the fact that M commutes with L_τ.

Now note that since $\|\nabla \tau\|_\infty \leq 1/2$, for any signal x we have

$$\|L_\tau x\|^2 = \int \|x(u - \tau(u))\|^2 du = \int \|x(u)\|^2 |\det(I - \nabla\tau(u))|^{-1} du$$
$$\leq \frac{1}{(1 - \|\nabla\tau\|_\infty)^d} \|x\|^2 \leq 2^d \|x\|^2. \tag{15}$$

Thus, we have $\|L_\tau y_n\| \leq 2^{d/2} \|y_n\| \leq \sqrt{2^d n} \|x\|$. Separately, using the non-expansivity of φ_1, we have

$$\|x_n^{(0)} - L_\tau x_n\| \leq \sum_{k=1}^{n-1} \|x_n^{(k-1)} - x_n^{(k)}\| + \|x_n^{(n-1)} - L_\tau x_n\|$$
$$\leq \sum_{k=1}^{n} \|x_k^{(k-1)} - L_\tau x_k\|$$
$$= \|P^1 A^0 L_\tau x - L_\tau P^1 A^0 x\| + \sum_{k=2}^{n} \|P^k A^{k-1} L_\tau \varphi_1(x_{k-1}) - L_\tau P^k A^{k-1} \varphi_1(x_{k-1})\|$$
$$\leq \sum_{k=1}^{n} \|[P^k A^{k-1}, L_\tau]\| \|x\|$$
$$\leq C(\beta) n \|\nabla\tau\|_\infty \|x\|,$$

by Lemma 15. We also have

$$\|y_n^{(0)} - L_\tau y_n\| = \|P^n A^{n-1} M(x_{n-1}^{(0)}, y_{n-1}^{(0)}) - L_\tau P^n A^{n-1} M(x_{n-1}, y_{n-1})\|$$
$$\leq \|P^n A^{n-1} M(x_{n-1}^{(0)}, y_{n-1}^{(0)}) - P^n A^{n-1} L_\tau M(x_{n-1}, y_{n-1})\| + \|[P^n A^{n-1}, L_\tau]\| \|M(x_{n-1}, y_{n-1})\|$$
$$\leq \|A^{n-1} M(x_{n-1}^{(0)}, y_{n-1}^{(0)}) - A^{n-1} L_\tau M(x_{n-1}, y_{n-1})\| + C(\beta)\sqrt{n} \|\nabla\tau\|_\infty \|x\|.$$

We have thus shown:

$$\|A^n M(x_n^{(0)}, y_n^{(0)}) - A^n L_\tau M(x_n, y_n)\| \leq \left(2^{d/4} C(\beta)^{1/2} n^{3/4} \|\nabla\tau\|_\infty^{1/2} + C(\beta)(n + \sqrt{n}) \|\nabla\tau\|_\infty\right) \|x\|$$
$$+ \|A^{n-1} M(x_{n-1}^{(0)}, y_{n-1}^{(0)}) - A^{n-1} L_\tau M(x_{n-1}, y_{n-1})\|.$$

Unrolling the recurrence relation yields

$$\|A^n M(x_n^{(0)}, y_n^{(0)}) - A^n L_\tau M(x_n, y_n)\| \leq \sum_{k=2}^{n} \left(2^{d/4} C(\beta)^{1/2} k^{3/4} \|\nabla\tau\|_\infty^{1/2} + C(\beta)(k + \sqrt{k}) \|\nabla\tau\|_\infty\right) \|x\|$$
$$\leq \left(C(\beta)^{1/2} C n^{7/4} \|\nabla\tau\|_\infty^{1/2} + C(\beta) C' n^2 \|\nabla\tau\|_\infty\right) \|x\|,$$

where C, C' are absolute constants depending only on d.

The final bound becomes

$$\|\Phi_n(L_\tau x) - \Phi_n(x)\| \leq \left(C(\beta)^{1/2} C n^{7/4} \|\nabla \tau\|_\infty^{1/2} + C(\beta) C' n^2 \|\nabla \tau\|_\infty + \sqrt{n+1} \frac{C_2}{\sigma_n} \|\tau\|_\infty \right) \|x\|,$$

with a different constant C'.

$\square$

C   Approximation Properties

C.1   Background on spherical harmonics

In this section, we provide some background on spherical harmonic analysis needed for our study of the RKHS. See [22, 5] for references, as well as [6, Appendix D]. We consider inputs on the $p-1$ sphere $\mathbb{S}^{p-1} = \{x \in \mathbb{R}^p, \|x\| = 1\}$.

We denote by $Y_{kj}(x)$, $j = 1, \ldots, N(p, k)$, the spherical harmonics of degree k on $\mathbb{S}^{p-1}$, where $N(p, k) = \frac{2k+p-2}{k} \binom{k+p-3}{p-2}$. They form an orthonormal basis of $L^2(\mathbb{S}^{p-1}, d\tau)$, where τ is the uniform measure on the sphere. The index k plays the role of an integer frequency, as in Fourier series. We have the addition formula

$$\sum_{j=1}^{N(p,k)} Y_{k,j}(x) Y_{k,j}(y) = N(p, k) P_k(\langle x, y \rangle), \tag{16}$$

where P_k is the k-th Legendre polynomial in dimension p (also known as Gegenbauer polynomials), given by the Rodrigues formula:

$$P_k(t) = (-1/2)^k \frac{\Gamma(\frac{p-1}{2})}{\Gamma(k + \frac{p-1}{2})} (1 - t^2)^{(3-p)/2} \left(\frac{d}{dt} \right)^k (1 - t^2)^{k+(d-3)/2}.$$

The polynomials P_k are orthogonal in $L^2([-1, 1], d\nu)$ where the measure $d\nu$ is given by $d\nu(t) = (1 - t^2)^{(p-3)/2} dt$, and we have

$$\int_{-1}^{1} P_k^2(t)(1 - t^2)^{(p-3)/2} dt = \frac{\omega_{p-1}}{\omega_{p-2}} \frac{1}{N(p, k)}, \tag{17}$$

where $\omega_{d-1} = \frac{2\pi^{d/2}}{\Gamma(d/2)}$ denotes the surface of the sphere $\mathbb{S}^{d-1}$ in d dimensions. Using the addition formula (16) and orthogonality of spherical harmonics, we can show

$$\int P_j(\langle w, x \rangle) P_k(\langle w, y \rangle) d\tau(w) = \frac{\delta_{jk}}{N(p, k)} P_k(\langle x, y \rangle) \tag{18}$$

Further, we have the recurrence relation [22, Eq. 4.36]

$$tP_k(t) = \frac{k}{2k + p - 2} P_{k-1}(t) + \frac{k + p - 2}{2k + p - 2} P_{k+1}(t), \tag{19}$$

for $k \geq 1$, and for $k = 0$ we simply have $tP_0(t) = P_1(t)$.

The Funk-Hecke formula is helpful for computing Fourier coefficients in the basis of spherical harmonics in terms of Legendre polynomials: for any $j = 1, \ldots, N(p, k)$, we have

$$\int f(\langle x, y \rangle) Y_{k,j}(y) d\tau(y) = \frac{\omega_{p-2}}{\omega_{p-1}} Y_{k,j}(x) \int_{-1}^{1} f(t) P_k(t)(1 - t^2)^{(p-3)/2} dt. \tag{20}$$

C.2 Dot-product kernels and spherical harmonics

In this section, we provide some background on how spherical harmonics can be used for obtaining descriptions of the RKHS for dot-product kernels on the $p-1$ sphere [46, 47]. We then recall results from Bach [6] on such a description for kernels arising from positively-homogeneous activations (*i.e.*, arc-cosine kernels), and on how differentiability of functions on the sphere relates to the RKHS.

For a positive-definite kernel $K(x,y) = \kappa(\langle x,y \rangle)$ defined on the $p-1$ sphere $\mathbb{S}^{p-1}$, we have the following Mercer decomposition:

$$\kappa(\langle x,y \rangle) = \sum_{k=0}^{\infty} \mu_k \sum_{j=1}^{N(p,k)} Y_{k,j}(x)Y_{k,j}(y) = \sum_{k=0}^{\infty} \mu_k N(p,k)P_k(\langle x,y \rangle), \tag{21}$$

where μ_k is obtained by computing Fourier coefficients of $\kappa(\langle x, \cdot \rangle)$ using (20):

$$\mu_k = \frac{\omega_{p-2}}{\omega_{p-1}} \int_{-1}^{1} \kappa(t)P_k(t)(1-t^2)^{(p-3)/2}dt. \tag{22}$$

Note that we must have $\mu_k \geq 0$ for all k, since positive definiteness of k would be violated if this were not true. Then, the RKHS $\mathcal{H}$ consists of functions of the form

$$f(x) = \sum_{k=0}^{\infty} \sum_{j=1}^{N(p,k)} a_{k,j}Y_{k,j}(x), \tag{23}$$

$$\text{s.t.} \quad \|f\|_{\mathcal{H}}^2 = \sum_{k=0}^{\infty} \sum_{j=1}^{N(p,k)} \frac{a_{k,j}^2}{\mu_k} < \infty. \tag{24}$$

In particular, this requires that $a_{k,j} = 0$ for all j, for any k such that $\mu_k = 0$.

Relationship with differentiability. Following Bach [6, Appendix D.3], if f is s-times differentiable with derivatives bounded by η, then we have $\|(-\Delta)^{s/2}f\|_{L^2(\mathbb{S}^{p-1})} \leq \eta$, where Δ is the Laplace-Beltrami operator on the sphere [5]. Given a function f as in (23), we can use the fact that $Y_{k,j}$ is an eigenfunction of $-\Delta$ with eigenvalue $k(k+p-2)$ (see [5, 22]) and write $a_{k,j} = a'_{k,j}/(k(k+p-2))^{s/2}$ for $k \geq 1$, where $a'_{k,j}$ are the Fourier coefficients of $(-\Delta)^{s/2}f$ and satisfy $\sum_{k,j} a_{k,j}'^2 \leq \eta^2$. We then have

Lemma 16. *Assume f takes the form* (23), *with $a_{k,j} = 0$ for all j when $\mu_k = 0$, and that f is s-times differentiable with derivatives bounded by η. Then, if $\max_{k \geq 1, \mu_k \neq 0} 1/(k^{2s}\mu_k) < C$, we have $f \in \mathcal{H}$ with $\|f\|_{\mathcal{H}}^2 \leq C'\eta^2$, with $C' = 1/\mu_0 + C$ if $\mu_0 \neq 0$, or $C' = C$ if $\mu_0 = 0$.*

Indeed, under the stated conditions, we have

$$\|f\|_{\mathcal{H}}^2 = \sum_{k=0}^{\infty} \sum_{j=1}^{N(p,k)} \frac{a_{k,j}^2}{\mu_k} \leq \frac{a_{0,1}^2}{\mu_0} + \max_{k \geq 1, \mu_k \neq 0} 1/(k^{2s}\mu_k) \sum_{k,j} a_{k,j}'^2$$

$$\leq \frac{1}{\mu_0}\eta^2 + C\|(-\Delta)^{s/2}f\|_{L^2(\mathbb{S})}^2 \leq C'\eta^2.$$

Values of μ_k for arc-cosine 0/1 kernels [6]. We recall values of the eigenvalues μ_k for arc-cosine kernels of degree 0 and 1, which are obtained in [6]. For any non-negative integer $\alpha \geq 0$, Bach [6] considers the positively homogeneous activation $\sigma_\alpha(u) = (u)_+^\alpha$ and derives the following quantities for $k \geq 0$:

$$\lambda_{\alpha,k} = \frac{\omega_{p-2}}{\omega_{p-1}} \int_{-1}^{1} \sigma_\alpha(t)P_k(t)(1-t^2)^{(p-3)/2}dt.$$

This can be used to derive the decompositions of the arc-cosine kernels introduced in Section 2, which are defined using expectations on Gaussian variables, but can be expressed using expectations on the sphere as follows:

$$\kappa_\alpha(\langle x,y \rangle) = 2\,\mathbb{E}_{w \sim \mathcal{N}(0,1)}[\sigma_\alpha(\langle w,x \rangle)\sigma_\alpha(\langle w,y \rangle)]$$

$$= 2\,\mathbb{E}_{w \sim \mathcal{N}(0,1)}[\|w\|^{2\alpha}\sigma_\alpha(\langle w/\|w\|,x \rangle)\sigma_\alpha(\langle w/\|w\|,y \rangle)]$$

$$= 2\,\mathbb{E}_{w \sim \mathcal{N}(0,1)}[\|w\|^{2\alpha}] \int \sigma_\alpha(\langle w,x \rangle)\sigma_\alpha(\langle w,y \rangle)d\tau(w),$$

where we used α-homogeneity of σ_α and the rotational symmetry of the normal distribution, which implies that $w/\|w\|$ is uniformly distributed on the sphere, and independent from $\|w\|$.

For a fixed w, we can express

$$\sigma_\alpha(\langle w, x \rangle) = \sum_{k=0}^{\infty} \lambda_{\alpha,k} N(p,k) P_k(\langle w, y \rangle).$$

Then, using (18), we have

$$\kappa_\alpha(\langle x, y \rangle) = 2\, \mathbb{E}_{w \sim \mathcal{N}(0,1)}[\|w\|^{2\alpha}] \sum_{k=0}^{\infty} \lambda_{\alpha,k}^2 N(p,k) P_k(\langle x, y \rangle).$$

For $\alpha = 0, 1$, this yields decompositions (21) of κ_0, κ_1 with $\mu_{0,k} = 2\lambda_{0,k}^2$ and $\mu_{1,k} = 2p\lambda_{1,k}^2$. Bach [6] then shows the following result on the decomposition of σ_α, which we translate to decompositions of kernel functions κ_α.

Lemma 17 (Decomposition of σ_α and κ_α [6]). *For the activation σ_α on the $p-1$ sphere, we have*

- $\lambda_{\alpha,k} \neq 0$ if $k \leq \alpha$;
- $\lambda_{\alpha,k} = 0$ if $k > \alpha$ if $k = \alpha \mod 2$;
- $|\lambda_{\alpha,k}| \sim C_\lambda(p, \alpha) k^{-p/2 - \alpha}$ *otherwise, for some constant $C_\lambda(p, \alpha)$ depending on p and α.*

For $\alpha \in \{0, 1\}$, the eigenvalues for the corresponding kernel κ_α then satisfy

- $\mu_{\alpha,k} > 0$ if $k \leq \alpha$;
- $\mu_{\alpha,k} = 0$ if $k > \alpha$ if $k = \alpha \mod 2$;
- $\mu_{\alpha,k} \sim C_\mu(p, \alpha) k^{-p - 2\alpha}$ *otherwise, with $C_\mu(p, \alpha) = 2p^\alpha C_\lambda(p, \alpha)^2$.*

Note that the zero eigenvalues imply that a function f of the form (23) must have $a_{k,j} = 0$ for $k > \alpha$ and $k = \alpha \mod 2$ in order to be in the RKHS for κ_α. A sufficient condition for this to hold is that f is even (resp. odd) when α is odd (resp. even) [6].

C.3 Decomposition of the NTK for two-layer ReLU networks

Recall that the NTK for the two-layer ReLU network with inputs on the sphere is given by

$$\kappa(u) = u\kappa_0(u) + \kappa_1(u).$$

We now prove Proposition 5, which shows that the Mercer decomposition in spherical harmonics (21) for this NTK satisfies:

- $\mu_0, \mu_1 > 0$
- $\mu_k = 0$ if $k = 2j + 1, j \geq 1$
- $\mu_k \sim C(p) k^{-p}$ otherwise, with $C(p) = C_\mu(p, 0)$.

Proof of Proposition 5. Using (22) and the recurrence relation (19) for Legendre polynomials, as well as $tP_0(t) = P_1(t)$, we have

$$\mu_0 = \mu_{0,1} + \mu_{1,0}$$
$$\mu_k = \frac{k}{2k + p - 2}\mu_{0,k-1} + \frac{k + p - 2}{2k + p - 2}\mu_{0,k+1} + \mu_{1,k}, \quad \text{for } k \geq 1.$$

By Lemma 17, we have the desired properties. $\qquad\square$

We now briefly discuss how to adapt the approximation results of Bach [6] to our setting.

Proof sketch of Corollaries 6 and 7. In Appendix D of [6], Bach defines candidate functions $g : \mathbb{S}^{p-1}$ from functions $p \in L^2(\mathbb{S}^{p-1})$ as $g(x) = Tp(x) := \int p(w)\sigma_\alpha(w^\top x)d\tau(w)$, with RKHS norm (denoted $\gamma_2(g)$ in [6]) given by the smallest $\|p\|_{L^2}$ for p such that $g = Tp$. In our case with the NTK, we may simply consider the operator $\Sigma^{1/2}$ instead (the self-adjoint square root of the integral operator Σ of κ, using notations from [7]; see also [18]), which simply multiplies each fourier coefficient $a_{k,j}$ in decomposition (23) by $\sqrt{\mu_k}$, and obeys the required properties (in fact, T and $\Sigma^{1/2}$ are two different square roots of Σ [7]).

The proofs can then be adapted directly, by noticing that $\sqrt{\mu_k}$ has the same decay properties as $\lambda_{\alpha,k}$ with $\alpha = 0$. For Corollary 6, we also provide the key proof ingredients in our framework in Lemma 16 for completeness. $\qquad\square$

The proof for the homogeneous RKHS is given below.

Proof of Proposition 8. The kernel K can be written as

$$K(x, x') = \langle \|x\|\Phi\left(\frac{x}{\|x\|}\right), \|x'\|\Phi\left(\frac{x'}{\|x'\|}\right)\rangle_{\mathcal{H}},$$

where $\Phi(\cdot)$ is the kernel mapping of the kernel κ on the sphere. Then the RKHS $\bar{\mathcal{H}}$ can be characterized by the following classical result (see, *e.g.*, [44, §2.1] or [6, Appendix A]):

$$\bar{\mathcal{H}} = \underbrace{\{x \mapsto \langle g, \|x\|\Phi\left(\frac{x}{\|x\|}\right)\rangle_{\mathcal{H}} : g \in \mathcal{H}\}}_{=:f_g}$$

$$\|f_g\|_{\bar{\mathcal{H}}} = \inf\{\|g'\|_{\mathcal{H}} : g' \in \mathcal{H} \text{ s.t. } f_g = f_{g'}\}.$$

Note that the condition $f_g = f_{g'}$ implies in particular that f_g and $f_{g'}$ are equal on the sphere, and thus that $g = g'$, so that the infimum is simply equal to $\|g\|_{\mathcal{H}}$. This concludes the proof.

$\qquad\square$

D Details on Numerical Experiments

This section provides more details on the experimental setup used for obtaining Figure 1, which closely follows the setup in [11, Section 3.4].

We consider images of handwritten digits from the Infinite MNIST dataset [32], which consists of 28x28 grayscale MNIST digits augmented with small translations and deformations. Translations are chosen at random from one of eight possible directions, while deformations are generated by considering small smooth deformations τ, and approximating $L_\tau x$ using a tangent vector field ∇x containing partial derivatives of the signal x along the horizontal and vertical image directions. We introduce a deformation parameter α to control deformation size. The images are then given by

$$L_{\alpha\tau}x(u) = x(u - \alpha\tau(u)) \approx x(u) - \alpha\tau(u) \cdot \nabla x(u).$$

Figure 2 shows examples of different deformations, with various values of α, with or without translations, generated from a reference image of the digit "5". In addition, one may consider that a given reference image of a handwritten digit can be deformed into different images of the same digit, and perhaps even into a different digit (*e.g.*, a "1" may be deformed into a "7"). Intuitively, the latter transformation corresponds to a "larger" deformation than the former, so that a prediction function that is stable to deformations should be preferable for a classification task.

The architecture underlying the kernels considered in Figure 1 consists of 2 convolutional layers with patches of size 3x3, followed by ReLU activations, and Gaussian pooling layers adapted to subsampling factors 2 for the first layer and 5 for the second. Patch extraction is performed with zero padding in order to preserve the size of the previous feature map. For a subsampling factor s, we apply a normalized Gaussian filter with scale $\sigma = s/\sqrt{2}$ and size $(2s+1) \times (2s+1)$, before downsampling. Our C++ implementation for computing the full kernel given two images is based on dynamic programming and is available at `https://github.com/albietz/ckn_kernel`.
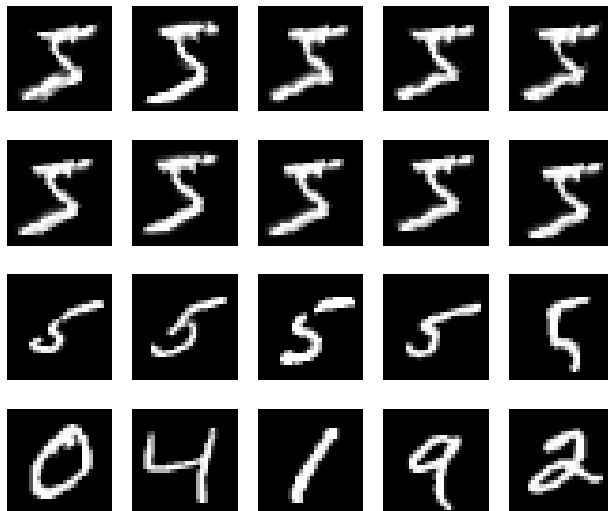
Figure 2: MNIST digits with transformations considered in our numerical study of stability. Each row gives examples of images from a set of digits that are compared to a reference image of a "5". From top to bottom: deformations with $\alpha = 3$; translations and deformations with $\alpha = 1$; digits from the training set with the same label "5" as the reference digit; digits from the training set with any label.

The plots in Figure 1 show average relative distance in the RKHS between a reference image and images from various sets of 20 images (either generated transformations, or images appearing in the training set). The figures consider deformations of varying size $\alpha \in \{0.01, 0.03, 0.1, 0.3, 1, 3\}$. For a given kernel K and set S of images, the average relative distance to an image x is given by

$$\frac{1}{|S|} \sum_{x' \in S} \frac{\|\Phi(x') - \Phi(x)\|_{\mathcal{H}}}{\|\Phi(x)\|_{\mathcal{H}}} = \frac{1}{|S|} \sum_{x' \in S} \frac{\sqrt{K(x,x) + K(x',x') - 2K(x,x')}}{\sqrt{K(x,x)}},$$

where $\mathcal{H}$ denotes the RKHS associated to K and Φ the kernel mapping. We normalize by $\|\Phi(x)\|_{\mathcal{H}}$ in order to reduce sensitivity to the choice of kernel.