A FINE-GRAINED SPECTRAL PERSPECTIVE ON NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Are neural networks biased toward simple functions? Does depth always help learn more complex features? Is training the last layer of a network as good as training all layers? These questions seem unrelated at face value, but in this work we give all of them a common treatment from the spectral perspective. We will study the spectra of the *Conjugate Kernel, CK,* (also called the *Neural Network-Gaussian Process Kernel*), and the *Neural Tangent Kernel, NTK*. Roughly, the CK and the NTK tell us respectively "what a network looks like at initialization" and "what a network looks like during and after training." Their spectra then encode valuable information about the initial distribution and the training and generalization properties of neural networks. By analyzing the eigenvalues, we lend novel insights into the questions put forth at the beginning, and we verify these insights by extensive experiments of neural networks. We believe the computational tools we develop here for analyzing the spectra of CK and NTK serve as a solid foundation for future studies of deep neural networks. We have open-sourced the code for it and for generating the plots in this paper at `github.com/jxVmnLgedVwv6mNcGCBy/NNspectra`.

1 INTRODUCTION

Understanding the behavior of neural networks and why they generalize has been a central pursuit of the theoretical deep learning community. Recently, Valle-Pérez et al. (2018) observed that neural networks have a certain "simplicity bias" and proposed this as a solution to the generalization question. One of the ways with which they argued that this bias exists is the following experiment: they drew a large sample of boolean functions by randomly initializing neural networks and thresholding the output. They observed that there is a bias toward some "simple" functions which get sampled disproportionately more often. However, their experiments were only done for relu networks. Can one expect this "simplicity bias" to hold universally, for any architecture?

A priori, this seems difficult, as the nonlinear nature seems to present an obstacle in reasoning about the distribution of random networks. However, this question turns out to be more easily treated if we allow the *width to go to infinity*. A long line of works starting with Neal (1995) and extended recently by Lee et al. (2018); Novak et al. (2018); Yang (2019) have shown that randomly initialized, infinite-width networks are distributed as Gaussian processes. These Gaussian processes also describe finite width random networks well (Valle-Pérez et al., 2018). We will refer to the corresponding kernels as the *Conjugate Kernels* (CK), following the terminology of Daniely et al. (2016). Given the CK K, the simplicity bias of a wide neural network can be read off quickly from the *spectrum of* K: If the largest eigenvalue of K accounts for most of $\operatorname{tr} K$, then a typical random network looks like a function from the top eigenspace of K.

In this paper, we will use this spectral perspective to probe not only the simplicity bias, but more generally, questions regarding how hyperparameters affect the generalization of neural networks.

Via the usual connection between Gaussian processes and linear models with features, the CK can be thought of as the kernel matrix associated to training only the last layer of a wide randomly initialized network. It is a remarkable recent advance (Jacot et al., 2018; Allen-Zhu et al., 2018a;c; Du et al., 2018) that, under a certain regime, a wide neural network of any depth evolves like a linear model even when training all parameters. The associated kernel is call the *Neural Tangent Kernel*, which is typically different from CK. While its theory was initially derived in the infinite width setting, Lee et al. (2019) confirmed with extensive experiment that this limit is predictive of finite width neural networks as well. Thus, just as the CK reveals information about *what a network looks like at initialization*, NTK reveals information about *what a network looks like after training.* As such, if

we can understand how hyperparameters change the NTK, we can also hope to understand how they affect the performance of the corresponding finite-width network.

Our Contributions In this paper, in addition to showing that the simplicity bias is not universal, we will attempt a first step at understanding the effects of the hyperparameters on generalization from a spectral perspective.

At the foundation is a spectral theory of the CK and the NTK on the boolean cube. In Section 3, we show that these kernels, as *integral operators* on functions over the boolean cube, are diagonalized by the natural Fourier basis, echoing similar results for over the sphere (Smola et al., 2001). We also partially diagonalize the kernels over standard Gaussian, and show that, as expected, the kernels over the different distributions (boolean cube, sphere, standard Gaussian) behave very similarly in high dimensions. However, the spectrum is much easier to compute over the boolean cube: while the sphere and Gaussian eigenvalues would require integration against a kind of polynomials known as the Gegenbauer polynomials, the boolean ones only require calculating a linear combination of a small number of terms. For this reason, in the rest of the paper we focus on analyzing the eigenvalues over the boolean cube.

Just as the usual Fourier basis over $\mathbb{R}$ has a notion of frequency that can be interpreted as a measure of complexity, so does the boolean Fourier basis (this is just the *degree*; see Section 3.1). While not perfect, we adopt this natural notion of complexity in this work; a "simple" function is then one that is well approximated by "low frequencies."

This spectral perspective immediately yields that the simplicity bias is not universal (Section 4). In particular, while it seems to hold more or less for relu networks, for sigmoidal networks, the simplicity bias can be made arbitrarily weak by changing the weight variance and the depth. In the extreme case, the random function obtained from sampling a deep erf network with large weights is distributed like a "white noise." However, there is a very weak sense in which the simplicity bias *does* hold: the eigenvalues of more "complex" eigenspaces cannot be bigger than those of less "complex" eigenspaces (Thm 4.1).

Next, we examine how hyperparameters affect the performance of neural networks through the lens of NTK and its spectrum. To do so, we first need to understand the simpler question of how a kernel affects the accuracy of the function learned by kernel regression. A coarse-grained theory, concerned with big-O asymptotics, exists from classical kernel literature (Yao et al., 2007; Raskutti et al., 2013; Wei et al.; Lin and Rosasco; Schölkopf and Smola, 2002). However, the fine-grained details, required for discerning the effect of hyperparameters, have been much less studied. We make a first attempt at a heuristic, *fractional variance* (i.e. what fraction of the trace of the kernel does an eigenspace contribute), for understanding how a minute change in kernel effects a change in performance. Intuitively, if an eigenspace has very large fractional variance, so that it accounts for most of the trace, then a ground truth function from this eigenspace should be very easy to learn.

Using this heuristic, we make two predictions about neural networks, motivated by observations in the spectra of NTK and CK, and verify them with extensive experiments.

- Deeper networks learn more complex features, but excess depth can be detrimental as well. Spectrally, depth can increase fractional variance of an eigenspace, but past an *optimal depth*, it will also decrease it. (Section 5) Thus, deeper is *not* always better.
- Training all layers is better than training just the last layer when it comes to more complex features, but the opposite is true for simpler features. Spectrally, fractional variances of more "complex" eigenspaces for the NTK are larger than the correponding quantities of the CK. (Section 6)

We emphasize that our theory and experiments only concern data distributed uniformly over the boolean cube, the sphere, or as a standard Gaussian. These distributions are theoretically interesting but insights found over them do not always carry over to real datasets. We leave for future work the extension to more practically relevant data distributions [1].

For space concerns, we review relevant literature along the flow of the main text, and relegate a more complete discussion of the related research landscape in Appendix A.

[1] The code for computing the eigenvalues and for reproducing the plots of this paper is available at `github.com/jxVmnLgedVwv6mNcGCBy/NNspectra`, which will be open sourced upon publication.

2 KERNELS ASSOCIATED TO NEURAL NETWORKS

As mentioned in the introduction, we now know several kernels associated to infinite width, randomly initialized neural networks. The most prominent of these are the *neural tangent kernel* (NTK) (Jacot et al., 2018) and the *conjugate kernel* (CK) (Daniely et al., 2016), which is also called the *NNGP kernel* (Lee et al., 2018). We briefly review them below. First we introduce the following notation that we will repeatedly use.

Definition 2.1. For $\phi : \mathbb{R} \to \mathbb{R}$, write V_ϕ for the function that takes a PSD (positive semidefinite) kernel function to a PSD kernel of the same domain by the formula

$$\mathrm{V}_\phi(K)(x, x') = \mathop{\mathbb{E}}_{f \sim \mathcal{N}(0,K)} \phi(f(x))\phi(f(x')).$$

Conjugate Kernel Neural networks are commonly thought of as learning a high-quality embedding of inputs to the latent space represented by the network's last hidden layer, and then using its final linear layer to read out a classification given the embedding. The conjugate kernel is just the kernel associated to the embedding induced by a random initialization of the neural network. Consider an MLP with widths $\{n^l\}_l$, weight matrices $\{W^l \in \mathbb{R}^{n^l \times n^{l-1}}\}_l$, and biases $\{b^l \in \mathbb{R}^{n^l}\}_l$, $l = 1, \ldots, L$. For simplicity of exposition, in this paper, we will only consider scalar output $n^L = 1$. Suppose it is parametrized by the *NTK parametrization*, i.e. its computation is given recursively as

$$h^1(x) = \frac{\sigma_w}{\sqrt{n^0}} W^1 x + \sigma_b b^1 \quad \text{and} \quad h^l(x) = \frac{\sigma_w}{\sqrt{n^0}} W^l \phi(h^{l-1}(x)) + \sigma_b b^l \tag{MLP}$$

with some hyperparameters σ_w, σ_b that are fixed throughout training[2]. At initialization time, suppose $W^l_{\alpha\beta}, b^l_\alpha \sim \mathcal{N}(0, 1)$ for each $\alpha \in [n^l], \beta \in [n^{l-1}]$. It can be shown that, for each $\alpha \in [n^l]$, h^l_α is a Gaussian process with zero mean and kernel function Σ^l in the limit as all hidden layers become infinitely wide ($n^l \to \infty, l = 1, \ldots, L-1$), where Σ^l is defined inductively on l as

$$\Sigma^1(x, x') \stackrel{\text{def}}{=} \sigma_w^2 (n^0)^{-1} \langle x, x' \rangle + \sigma_b^2, \quad \Sigma^l \stackrel{\text{def}}{=} \sigma_w^2 \mathrm{V}_\phi(\Sigma^{l-1}) + \sigma_b^2 \tag{CK}$$

The kernel Σ^L corresponding the the last layer L is the network's *conjugate kernel*, and the associated Gaussian process limit is the reason for its alternative name *Neural Network-Gaussian process kernel*. In short, if we were to train a linear model with features given by the embedding $x \mapsto h^{L-1}(x)$ when the network parameters are randomly sampled as above, then the CK is the kernel of this linear model. See Daniely et al. (2016); Lee et al. (2018) and Appendix D for more details.

Neural Tangent Kernel On the other hand, the NTK corresponds to training the entire model instead of just the last layer. Intuitively, if we let θ be the entire set of parameters $\{W^l\}_l \cup \{b^l\}_l$ of Eq. (MLP), then for θ close to its initialized value θ_0, we expect

$$h^L(x; \theta) - h^L(x; \theta_0) \approx \langle \nabla_\theta h^L(x; \theta_0), \theta - \theta_0 \rangle$$

via a naive first-order Taylor expansion. In other words, $h^L(x; \theta) - h^L(x; \theta_0)$ behaves like a linear model with feature of x given by the gradient taken w.r.t. the initial network, $\nabla_\theta h^L(x; \theta_0)$, and the weights of this *linear model* are the deviation $\theta - \theta_0$ of θ from its initial value. It turns out that, in the limit as all hidden layer widths tend to infinity, this intuition is correct (Jacot et al., 2018; Lee et al., 2018; Yang, 2019), and the following inductive formula computes the corresponding infinite-width kernel of this linear model:

$$\Theta^1 \stackrel{\text{def}}{=} \Sigma^1, \quad \Theta^l(x, x') \stackrel{\text{def}}{=} \Sigma^l(x, x') + \sigma_w^2 \Theta^{l-1}(x, x') \mathrm{V}_{\phi'}(\Sigma^{l-1})(x, x'). \tag{NTK}$$

Computing CK and NTK While in general, computing V_ϕ and $\mathrm{V}_{\phi'}$ requires evaluating a multivariate Gaussian expectation, in specific cases, such as when $\phi = \mathrm{relu}$ or erf, there exists explicit, efficient formulas that only require pointwise evaluation of some simple functions (see Facts D.1 and D.2). This allows us to evaluate CK and NTK on a set $\mathcal{X}$ of inputs in only time $O(|\mathcal{X}|^2 L)$.

[2] SGD with learning rate α in this parametrization is roughly equivalent to SGD with learning rate $\alpha/width$ in the standard parametrization with Glorot initialization; see Lee et al. (2018)

What Do the Spectra of CK and NTK Tell Us? In summary, the CK governs the distribution of a randomly initialized neural network and also the properties of training only the last layer of a network, while the NTK governs the dynamics of training (all parameters of) a neural network. A study of their spectra thus informs us of the "implicit prior" of a randomly initialized neural network as well as the "implicit bias" of GD in the context of training neural networks.

In regards to the implicit prior at initialization, we know from Lee et al. (2018) that a randomly initialized network as in Eq. (MLP) is distributed as a Gaussian process $\mathcal{N}(0, K)$, where K is the corresponding CK, in the infinite-width limit. If we have the eigendecomposition

$$K = \sum_{i \geq 1} \lambda_i u_i \otimes u_i \tag{1}$$

with eigenvalues λ_i in decreasing order and corresponding eigenfunctions u_i, then each sample from this GP can be obtained as

$$\sum_{i \geq 1} \sqrt{\lambda_i} \omega_i u_i, \quad \omega_i \sim \mathcal{N}(0, 1).$$

If, for example, $\lambda_1 \gg \sum_{i \geq 2} \lambda_i$, then a typical sample function is just a very small perturbation of u_1. We will see that for relu, this is indeed the case (Section 4), and this explains the "simplicity bias" in relu networks found by Valle-Pérez et al. (2018).

Training the last layer of a randomly initialized network via full batch gradient descent for an infinite amount of time corresponds to Gaussian process inference with kernel K (Lee et al., 2018; 2019). A similar intuition holds for NTK: training all parameters of the network (Eq. (MLP)) for an infinite amount of time yields the mean prediction of the GP $\mathcal{N}(0, \mathrm{NTK})$ in expectation; see Lee et al. (2019) and Appendix D.4 for more discussion.

Thus, the more the GP prior (governed by the CK or the NTK) is consistent with the ground truth function f^*, the more we expect the Gaussian process inference and GD training to generalize well. We can measure this consistency in the "alignment" between the eigenvalues λ_i and the squared coefficients a_i^2 of f^*'s expansion in the $\{u_i\}_i$ basis. The former can be interpreted as the expected magnitude (squared) of the u_i-component of a sample $f \sim \mathcal{N}(0, K)$, and the latter can be interpreted as the actual magnitude squared of such component of f^*. In this paper, we will investigate an even cleaner setting where $f^* = u_i$ is an eigenfunction. Thus we would hope to use a kernel whose ith eigenvalue λ_i is as large as possible.

Neural Kernels From the forms of the equation Eqs. (CK) and (NTK) and the fact that $\mathrm{V}_\phi(K)(x, x')$ only depends on $K(x, x), K(x, x')$, and $K(x', x')$, we see that CK or NTK of MLPs takes the form

$$K(x, y) = \Phi\left(\frac{\langle x, y \rangle}{\|x\|\|y\|}, \frac{\|x\|^2}{d}, \frac{\|y\|^2}{d}\right) \tag{2}$$

for some function $\Phi : \mathbb{R}^3 \to \mathbb{R}$. We will refer to this kind of kernel as *Neural Kernel* in this paper.

Kernels as Integral Operators We will consider input spaces of various forms $\mathcal{X} \subseteq \mathbb{R}^d$ equipped with some probability measure. Then a kernel function K acts as an *integral operator* on functions $f \in L^2(\mathcal{X})$ by

$$Kf(x) = (Kf)(x) = \mathop{\mathbb{E}}_{y \sim \mathcal{X}} K(x, y) f(y).$$

We will use the "juxtaposition syntax" Kf to denote this application of the integral operator. [3] Under certain assumptions, it then makes sense to speak of the eigenvalues and eigenfunctions of the integral operator K. While we will appeal to an intuitive understanding of eigenvalues and eigenfunctions in the main text below, we include a more formal discussion of Hilbert-Schmidt operators and their spectral theory in Appendix E for completeness. In the next section, we investigate the eigendecomposition of neural kernels as integral operators over different distributions.

[3] In cases when $\mathcal{X}$ is finite, K can be also thought of as a big matrix and f as a vector — but do not confuse Kf with their multiplication! If we use $\cdot$ to denote matrix multiplication, then the operator application Kf is the same as the matrix multiplication $K \cdot D \cdot f$ where D is the diagonal matrix encoding the probability values of each point in $\mathcal{X}$.

3 THE SPECTRA OF NEURAL KERNELS

3.1 BOOLEAN CUBE

We first consider a neural kernel K on the boolean cube $\mathcal{X} = \square^d \stackrel{\text{def}}{=} \{\pm 1\}^d$, equipped with the uniform measure. In this case, since each $x \in \mathcal{X}$ has the same norm, $K(x, y) = \Phi\left(\frac{\langle x,y\rangle}{\|x\|\|y\|}, \frac{\|x\|^2}{d}, \frac{\|y\|^2}{d}\right)$ effectively only depends on $\langle x, y\rangle$, so we will treat Φ as a single variate function in this section, $\Phi(c) = \Phi(c, 1, 1)$.

Brief review of basic Fourier analysis on the boolean cube $\square^d$ (O'Donnell (2014)). The space of real functions on $\square^d$ forms a 2^d-dimensional space. Any such function has a *unique* expansion into a *multilinear polynomial* (polynomials whose monomials do not contain $x_i^p, p \geq 2$, of any variable x_i). For example, the majority function over 3 bits has the following unique multilinear expansion

$$\text{maj}_3 : \square^3 \to \square^1, \quad \text{maj}_3(x_1, x_2, x_3) = \frac{1}{2}(x_1 + x_2 + x_3 - x_1x_2x_3).$$

In the language of Fourier analysis, the 2^d multilinear monomial functions

$$\chi_S(x) \stackrel{\text{def}}{=} x^S \stackrel{\text{def}}{=} \prod_{i \in S} x_i, \quad \text{for each } S \subseteq [d] \tag{3}$$

form a Fourier basis of the function space $L^2(\square^d) = \{f : \square^d \to \mathbb{R}\}$, in the sense that their inner products satisfy

$$\mathop{\mathbb{E}}_{x \sim \square^d} \chi_S(x)\chi_T(x) = \mathbb{I}(S = T).$$

It turns out that K is always diagonalized by this Fourier basis $\{\chi_S\}_{S \subseteq [d]}$.

Theorem 3.1. *On the d-dimensional boolean cube $\square^d$, for every $S \subseteq [d]$, χ_S is an eigenfunction of K with eigenvalue*

$$\mu_{|S|} \stackrel{\text{def}}{=} \mathop{\mathbb{E}}_{x \in \square^d} x^S K(x, \mathbb{1}) = \mathop{\mathbb{E}}_{x \in \square^d} x^S \Phi\left(\sum_i x_i/d\right), \tag{4}$$

where $\mathbb{1} = (1, \ldots, 1) \in \square^d$. This definition of $\mu_{|S|}$ does not depend on the choice S, only on the cardinality of S. These are all of the eigenfunctions of K by dimensionality considerations.[4]

Define $\mathcal{T}_\Delta$ to be the shift operator on functions over $[-1, 1]$ that sends $\Phi(\cdot)$ to $\Phi(\cdot - \Delta)$. Then we can re-express the eigenvalue as follows.

Lemma 3.2. *With μ_k as in Thm 3.1,*

$$\mu_k = 2^{-d}(I - \mathcal{T}_\Delta)^k (I + \mathcal{T}_\Delta)^{d-k}\Phi(1) \tag{5}$$

$$= 2^{-d} \sum_{r=0}^{d} C_r^{d-k,k} \Phi\left(\left(\frac{d}{2} - r\right)\Delta\right) \tag{6}$$

where

$$C_r^{d-k,k} \stackrel{\text{def}}{=} \sum_{j=0} (-1)^{r+j} \binom{d-k}{j}\binom{k}{r-j}. \tag{7}$$

Eq. (5) will be important for computational purposes, and we will come back to discuss this more in Section 3.5. It also turns out μ_k affords a pretty expression via the Fourier series coefficients of Φ. As this is not essential to the main text, we relegate its exposition to Appendix F.1.

[4] Readers familiar with boolean Fourier analysis may be reminded of the *noise operator* T_ρ, $\rho \leq 1$ (O'Donnell, 2014, Defn 2.46). In the language of this work, T_ρ is a neural kernel with eigenvalues $\mu_k = \rho^k$.

3.2 SPHERE

Now let's consider the case when $\mathcal{X} = \sqrt{d}\mathcal{S}^{d-1}$ is the radius-$\sqrt{d}$ sphere in $\mathbb{R}^d$ equipped with the uniform measure. Again, because $x \in \mathcal{X}$ all have the same norm, we will treat Φ as a univariate function with $K(x, y) = \Phi(\langle x, y\rangle/\|x\|\|y\|) = \Phi(\langle x, y\rangle/d)$. As is long known (Schoenberg, 1942; Gneiting, 2013; Xu and Cheney, 1992; Smola et al., 2001), K is diagonalized by spherical harmonics, and the eigenvalues are given by the coefficients of Φ against a system of orthogonal polynomials called Gegenbuaer polynomials. We relegate a complete review of this topic to Appendix F.2.

3.3 ISOTROPIC GAUSSIAN

Now let's consider $\mathcal{X} = \mathbb{R}^d$ equipped with standard isotropic Gaussian $\mathcal{N}(0, I)$, so that K behaves like

$$Kf(x) = \mathop{\mathbb{E}}_{y\sim\mathcal{N}(0,I)} K(x, y)f(y) = \mathop{\mathbb{E}}_{y\sim\mathcal{N}(0,I)} \Phi\left(\frac{\langle x, y\rangle}{\|x\|\|y\|}, \frac{\|x\|^2}{d}, \frac{\|y\|^2}{d}\right) f(y)$$

for any $f \in L^2(\mathcal{N}(0, I))$. In contrast to the previous two sections, K will essentially depend on the effect of the norms $\|x\|$ and $\|y\|$ on Φ.

Nevertheless, because an isotropic Gaussian vector can be obtained by sampling its direction uniformly from the sphere and its magnitude from a chi distribution, K can still be partially diagonalized into a sum of products between spherical harmonics and kernels on $\mathbb{R}$ equipped with a chi distribution (Thm F.11). In certain cases, we can obtain complete eigendecompositions, for example when Φ is positive homogeneous. See Appendix F.3 for more details.

3.4 KERNEL IS SAME OVER BOOLEAN CUBE, SPHERE, OR GAUSSIAN WHEN $d \gg 1$

The reason we have curtailed a detailed discussion of neural kernels on the sphere and on the standard Gaussian is because, in high dimension, the kernel behaves the same under these distributions as under uniform distribution over the boolean cube. Indeed, by intuition along the lines of the central limit theorem, we expect that uniform distribution over a high dimension boolean cube should approximate high dimensional standard Gaussian. Similarly, by concentration of measure, most of the mass of a Gaussian is concentrated around a thin shell of radius $\sqrt{d}$. Thus, morally, we expect the same kernel function K induces approximately the same integral operator on these three distributions in high dimension, and as such, their eigenvalues *should* also approximately coincide. We verify numerically this is indeed the case in Appendix F.4.

3.5 COMPUTING THE EIGENVALUES

As the eigenvalues of K over the different distributions are very close, we will focus in the rest of this paper on eigenvalues over the boolean cube. This has the additional benefit of being much easier to compute.

Each eigenvalue over the sphere and the standard Gaussian requires an integration of Φ against a Gegenbauer polynomial. In high dimension d, these Gegenbauer polynomials varies wildly in a sinusoidal fashion, and blows up toward the boundary (see Fig. 10 in the Appendix). As such, it is difficult to obtain a numerically stable estimate of this integral in an efficient manner when d is large.

In contrast, we have multiple ways of computing boolean cube eigenvalues, via Eqs. (5) and (6). In either case, we just take some linear combination of the values of Φ at a grid of points on $[-1, 1]$, spaced apart by $\Delta = 2/d$. While the coefficients $C_r^{d-k,k}$ (defined in Eq. (7)) are relatively efficient to compute, the change in the sign of $C_r^{d-k,k}$ makes this procedure numerically unstable for large d. Instead, we use Eq. (5) to isolate the alternating part to evaluate in a numerically stable way: Since $\mu_k = \left(\frac{I+\mathcal{T}_\Delta}{2}\right)^{d-k}\left(\frac{I-\mathcal{T}_\Delta}{2}\right)^k \Phi(1)$, we can evaluate $\tilde{\Phi} \stackrel{\text{def}}{=} \left(\frac{I-\mathcal{T}_\Delta}{2}\right)^k \Phi$ via k finite differences, and then compute

$$\left(\frac{I + \mathcal{T}_\Delta}{2}\right)^{d-k} \tilde{\Phi}(1) = \frac{1}{2^{d-k}} \sum_{r=0}^{d-k} \binom{d-k}{r} \tilde{\Phi}(1 - r\Delta). \tag{8}$$

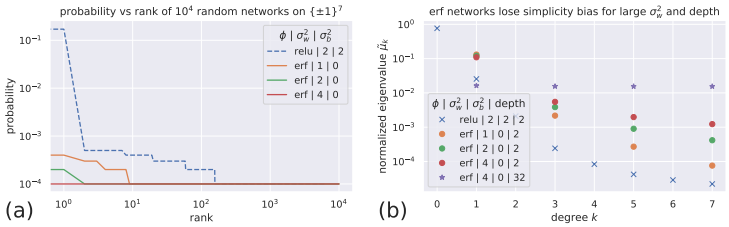


Figure 1: **The "simplicity bias" is not so simple.** **(a)** Following Valle-Pérez et al. (2018), we sample 10^4 boolean functions $\{\pm1\}^7 \to \{\pm1\}$ as follows: for each combination of nonlinearity, weight variance σ_w^2, and bias variance σ_b^2 (as used in Eq. (MLP)), we randomly initialize a network of 2 hidden layers, 40 neurons each. Then we threshold the function output to a boolean output, and obtain a boolean function sample. We repeat this for 10^4 random seeds to obtain all samples. Then we sort the samples according to frequency (this is the x-axis, *rank*), and plot their empirical probability (this is the y-axis, *probability*). The high values at the left of the relu curve indicates that a few functions get sampled repeatedly, while this is less true for erf. For erf and $\sigma_w^2 = 4$, no function got sampled more than once. **(b)** For different combination of nonlinearity, σ_w^2, σ_b^2, and depth, we study the eigenvalues of the corresponding CK. Each CK has 8 different eigenvalues $\mu_0, \ldots, \mu_7$ corresponding to homogeneous polynomials of degree $0, \ldots, 7$. We plot them in log scale against the degree. Note that for erf and $\sigma_b = 0$, the even degree μ_k vanish. See main text for explanations.

When Φ arises from the CK or the NTK of an MLP, all derivatives of Φ at 0 are nonnegative (Thm G.3). Thus intuitively, the finite difference $\tilde{\Phi}$ should be also all nonnegative, and this sum can be evaluated without worry about floating point errors from cancellation of large terms.

A slightly more clever way to improve the numerical stability when $2k \leq d$ is to note that

$$(I + \mathcal{T}_\Delta)^{d-k} (I - \mathcal{T}_\Delta)^k \Phi(1) = (I + \mathcal{T}_\Delta)^{d-2k} \left(I - \mathcal{T}_\Delta^2\right)^k \Phi(1) = (I + \mathcal{T}_\Delta)^{d-2k} (I - \mathcal{T}_{2\Delta})^k \Phi(1).$$

So an improved algorithm is to first compute the kth finite difference $(I - \mathcal{T}_{2\Delta})^k$ with the larger step size 2Δ, then compute the sum $(I + \mathcal{T}_\Delta)^{d-2k}$ as in Eq. (8).

4 Clarifying the "Simplicity Bias" of Random Neural Networks

As mentioned in the introduction, Valle-Pérez et al. (2018) claims that neural networks are *biased toward simple functions*. We show that this phenomenon depends crucially on the nonlinearity, the sampling variances, and the depth of the network. In Fig. 1(a), we have repeated their experiment for 10^4 random functions obtained by sampling relu neural networks with 2 hidden layers, 40 neurons each, following Valle-Pérez et al. (2018)'s architectural choices[5]. We also do the same for erf networks of the same depth and width, varying as well the sampling variances of the weights and biases, as shown in the legend. As discussed in Valle-Pérez et al. (2018), for relu, there is indeed this bias, where a single function gets sampled more than 10% of the time. However, for erf, as we increase σ_w^2, we see this bias disappear, and every function in the sample gets sampled only once.

This phenomenon can be explained by looking at the eigendecomposition of the CK, which is the Gaussian process kernel of the distribution of the random networks as their hidden widths tend to infinity. In Fig. 1(b), we plot the normalized eigenvalues $\{\mu_k / \sum_{i=0}^7 \binom{7}{i} \mu_i\}_{k=0}^7$ for the CKs corresponding to the networks sampled in Fig. 1(a). Immediately, we see that for relu and $\sigma_w^2 = \sigma_b^2 = 2$, the degree 0 eigenspace, corresponding to constant functions, accounts for more than 80% of the variance. This means that a typical infinite-width relu network of 2 layers is expected to be almost constant, and this should be even more true after we threshold the network to be a boolean function. On the other hand, for erf and $\sigma_b = 0$, the even degree μ_ks all vanish, and most of the variance comes from degree 1 components (i.e. linear functions). This concentration in degree 1 also lessens as σ_w^2 increases. But because this variance is spread across a dimension 7 eigenspace, we don't see duplicate function samples nearly as much as in the relu case. As σ_w increases, we also see the eigenvalues become more equally distributed, which corresponds to the flattening of the probability-vs-rank curve in Fig. 1(a). Finally, we observe that a 32-layer erf network with

[5] Valle-Pérez et al. (2018) actually performed their experiments over the $\{0,1\}^7$ cube, not the $\{\pm1\}^7$ cube we are using here. This does not affect our conclusion. See Appendix H for more discussion

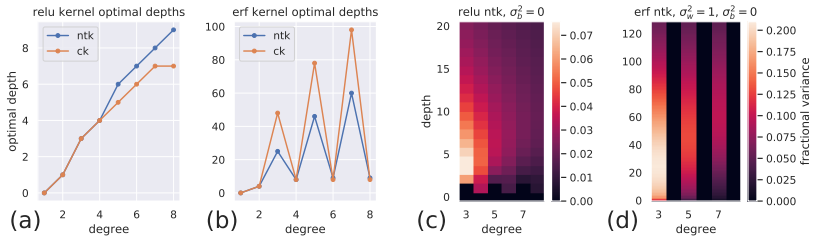


Figure 2: **The depth maximizing degree k fractional variance increases with k for both relu and erf.** For relu **(a)** and erf **(b)**, we plot for each degree k the depth such that there exists some combination of other hyperparameters (such as σ_b^2 or σ_w^2) that maximizes the degree k fractional variance. For both relu, $\sigma_b^2 = 0$ maximizes fractional variance in general, and same holds for erf in the odd degrees (see Appendix B), so we take a closer look at this slice by plotting heatmaps of fractional variance of various degrees versus depth for relu **(c)** and erf **(d)** NTK, with bright colors representing high variance. Clearly, we see the brightest region of each column, corresponding to a fixed degree, moves up as we increase the degree, barring for the even/odd degree alternating pattern for erf NTK. The pattern for CKs are similar and their plots are omitted.

$\sigma_w^2 = 4$ has all its nonzero eigenvalues (associated to odd degrees) all equal (see points marked by $*$ in Fig. 1(b)). This means that its distribution is a "white noise" on the space of *odd* functions, and the distribution of boolean functions obtained by thresholding the Gaussian process samples is the *uniform distribution* on *odd* functions. This is the complete lack of simplicity bias modulo the oddness constraint.

However, from the spectral perspective, there is a weak sense in which a simplicity bias holds for all neural network-induced CKs and NTKs.

Theorem 4.1 (Weak Spectral Simplicity Bias)**.** *Let K be the CK or NTK of an MLP on a boolean cube $\square^d$. Then the eigenvalues $\mu_k, k = 0, \ldots, d$, satisfy*

$$\mu_0 \geq \mu_2 \geq \cdots \geq \mu_{2k} \geq \cdots, \quad \mu_1 \geq \mu_3 \geq \cdots \geq \mu_{2k+1} \geq \cdots. \tag{9}$$

Even though it's not true that the fraction of variance contributed by the degree k eigenspace is decreasing with k, the eigenvalue themselves will be in a nonincreasing pattern across even and odd degrees. In fact, if we fix k and let $d \to \infty$, then we can show that (Thm G.6)

$$\mu_k = \Theta(d^{-k}).$$

Of course, as we have seen, this is a *very weak* sense of simplicity bias, as it doesn't prevent "white noise" behavior as in the case of erf CK with large σ_w^2 and large depth.

5 Deeper Networks Learn More Complex Features

In the rest of this work, we compute the eigenvalues μ_k over the 128-dimensional boolean cube ($\square^d$, with $d = 128$) for a large number of different hyperparameters, and analyze how the latter affect the former. We vary the degree $k \in [0, 8]$, the nonlinearity between relu and erf, the depth (number of hidden layers) from 1 to 128, and $\sigma_b^2 \in [0, 4]$. We fix $\sigma_w^2 = 2$ for relu kernels, but additionally vary $\sigma_w^2 \in [1, 5]$ for erf kernels. Comprehensive contour plots of how these hyperparameters affect the kernels are included in Appendix B, but in the main text we summarize several trends we see.

We will primarily measure the change in the spectrum by the *degree k fractional variance*, which is just

$$\text{degree } k \text{ fractional variance} \stackrel{\text{def}}{=} \frac{\binom{d}{k}\mu_k}{\sum_{i=0}^{d}\binom{d}{i}\mu_i}.$$

This terminology comes from the fact that, if we were to sample a function f from a Gaussian process with kernel K, then we expect that $r\%$ of the total variance of f comes from degree k components of f, where $r\%$ is the degree k fractional variance. If we were to try to learn a homogeneous degree-k polynomial using a kernel K, intuitively we should try to choose K such that its μ_k is maximized,

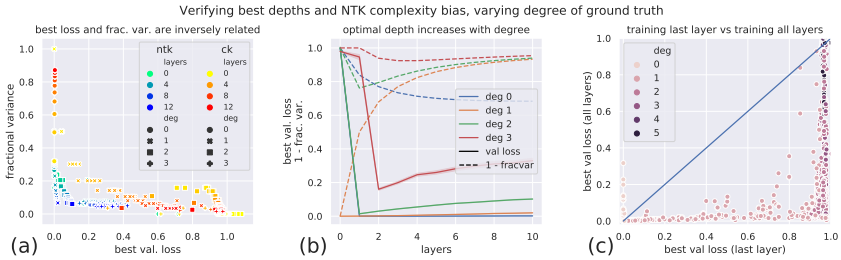


Figure 3: **(a)** We train relu networks of different depths against a ground truth polynomial on $\square^{128}$ of different degrees k. We either train only the last layer (marked "ck") or all layers (marked "ntk"), and plot the degree k fractional variance of the corresponding kernel against the best validation loss over the course of training. We see that the best validation loss is in general inversely correlated with fraction variance, as expected. However, their precise relationship seems to change depending on the degree, or whether training all layers or just the last. See Appendix C for experimental details. **(b)** Same experimental setting as (a), with slightly different hyperparameters, and plotting depth against best validation loss (solid curves), as well as the corresponding kernel's ($1-$ fractional variance) (dashed curves). We see that the loss-minimizing depth increases with the degree, as predicted by Fig. 2. Note that we do not expect the dashed and solid curves to match up, just that they are positively correlated as shown by (a). In higher degrees, the losses are high across all depths, and the variance is large, so we omit them. See Appendix C for experimental details. **(c)** Similar experimental setting as (a), but with more hyperparameters, and now comparing training last layer vs training all layers. The color of each dot indicates the degree of the ground truth polynomial. Below the identity line, training all layers is better than training last layer. We see that the only nontrivial case where this is not true is when learning degree 0 polynomials, i.e. constant functions. See Appendix C for experimental details.

relative to other eigenvalues. Fig. 3(a) shows that this is indeed the case even with neural networks: over a large number of different hyperparameter settings, degree k fractional variance is inversely related to the validation loss incurred when learning a degree k polynomial. However, this plot also shows that there does not seem like a precise, clean relationship between fractional variance and validation loss. Obtaining a better measure for predicting generalization is left for future work.

Before we continue, we remark that the fractional variance of a fixed degree k converges to a fixed value as the input dimension $d \to \infty$:

Theorem 5.1 (Asymptotic Fractional Variance)**.** *Let K be the CK or NTK of an MLP on a boolean cube $\square^d$. Then K can be expressed as $K(x, y) = \Phi(\langle x, y\rangle/d)$ for some analytic function $\Phi : \mathbb{R} \to \mathbb{R}$. If we fix k and let the input dimension $d \to \infty$, then the fractional variance of degree k converges to*

$$(k!)^{-1}\Phi^{(k)}(0)/\Phi(1) = \frac{(k!)^{-1}\Phi^{(k)}(0)}{\sum_{j\geq 0}(j!)^{-1}\Phi^{(j)}(0)}$$

where $\Phi^{(k)}$ denotes the kth derivative of Φ.

For the fractional variances we compute in this paper, their values at $d = 128$ are already very close to their $d \to \infty$ limit, so we focus on the $d = 128$ case experimentally.

If K were to be the CK or NTK of a relu or erf MLP, then we find that for higher k, the depth of the network helps increase the degree k fractional variance. In Fig. 2(a) and (b), we plot, for each degree k, the depth that (with some combination of other hyperparameters like σ_b^2) achieves this maximum, for respectively relu and erf kernels. Clearly, the maximizing depths are increasing with k for relu, and also for erf when considering either odd k or even k only. The slightly differing behavior between even and odd k is expected, as seen in the form of Thm 4.1. Note the different scales of y-axes for relu and erf — the depth effect is much stronger for erf than relu.

For relu NTK and CK, $\sigma_b^2 = 0$ maximizes fractional variance in general, and the same holds for erf NTK and CK in the odd degrees (see Appendix B). In Fig. 2(c) and Fig. 2(d) we give a more fine-grained look at the $\sigma_b^2 = 0$ slice, via heatmaps of fractional variance against degree and depth. Brighter color indicates higher variance, and we see the optimal depth for each degree k clearly increases with k for relu NTK, and likewise for odd degrees of erf NTK. However, note that as k increases, the difference between the maximal fractional variance and those slightly suboptimal

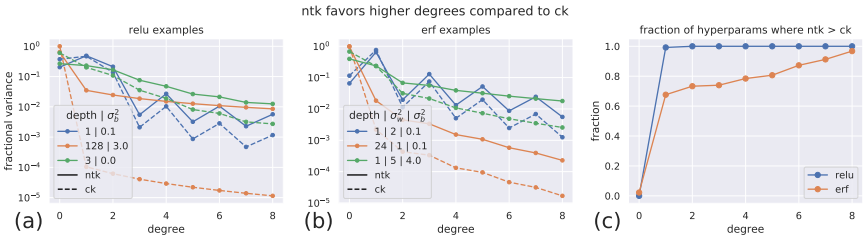


Figure 4: **Across nonlinearities and hyperparameters, NTK tends to have higher fraction of variance attributed to higher degrees than CK.** In **(a)**, we give several examples of the fractional variance curves for relu CK and NTK across several representative hyperparameters. In **(b)**, we do the same for erf CK and NTK. In both cases, we clearly see that, while for degree 0 or 1, the fractional variance is typically higher for CK, the reverse is true for larger degrees. In **(c)**, for each degree k, we plot the *fraction of hyperparameters* where the degree k fractional variance of NTK is greater than that of CK. Consistent with previous observations, this fraction increases with the degree.

becomes smaller and smaller, reflected by suppressed range of color moving to the right. The heatmaps for relu and erf CKs look similar and are omitted.

We verify this increase of optimal depth with degree in Fig. 3(b). There we have trained relu networks of varying depth against a ground truth multilinear polynomial of varying degree. We see clearly that the optimal depth is increasing with degree.

Note that implicit in our results here is a highly nontrivial observation: Past some point (the *optimal depth*), high depth can be detrimental to the performance of the network, beyond just the difficulty to train, and this detriment can already be seen in the corresponding NTK or CK. In particular, it's *not* true that the optimal depth is infinite. This adds significant nuance to the folk wisdom that "depth increases expressivity and allows neural networks to learn more complex features."

6 NTK Favors More Complex Features Than CK

We generally find the degree k fractional variance of NTK to be higher than that of CK when k is large, and vice versa when k is small, as shown in Fig. 4. This means that, if we train only the last layer of a neural network (i.e. CK dynamics), we intuitively should expect to learn simpler features better, while, if we train all parameters of the network (i.e. NTK dynamics), we should expect to learn more complex features better. Similarly, if we were to sample a function from a Gaussian process with the CK as kernel (recall this is just the distribution of randomly initialized infinite width MLPs (Lee et al., 2018)), this function is more likely to be accurately approximated by low degree polynomials than the same with the NTK.

We verify this intuition by training a large number of neural networks against ground truth functions of various homogeneous polynomials of different degrees, and show a scatterplot of how training the last layer only measures against training all layers (Fig. 3(c)). Consistent with our theory, the only place training the last layer works meaningfully better than training all layers is when the ground truth is a constant function. However, we reiterate that fractional variance is an imperfect indicator of performance. Even though for erf neural networks and $k \geq 1$, degree k fractional variance of NTK is not always greater than that of the CK, we do not see any instance where training the last layer of an erf network is better than training all layers. We leave an investigation of this discrepancy to future work.

7 Conclusion

In this work, we have taken a first step at studying how hyperparameters change the initial distribution and the generalization properties of neural networks through the lens of neural kernels and their spectra. We obtained interesting insights by computing kernel eigenvalues over the boolean cube and relating them to generalization through the *fractional variance* heuristic. While it inspired valid predictions that are backed up by experiments, fractional variance is clearly just a rough indicator. We hope future work can refine on this idea to produce a much more precise prediction of test loss. Nevertheless, we believe the spectral perspective is the right line of research that will not only shed light on mysteries in deep learning but also inform design choices in practice.

REFERENCES

Joshua Achiam, Ethan Knight, and Pieter Abbeel. Towards Characterizing Divergence in Deep Q-Learning. *arXiv:1903.08894 [cs]*, March 2019.

Zeyuan Allen-Zhu, Yuanzhi Li, and Yingyu Liang. Learning and Generalization in Overparameterized Neural Networks, Going Beyond Two Layers. *arXiv:1811.04918 [cs, math, stat]*, November 2018a.

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. On the Convergence Rate of Training Recurrent Neural Networks. *arXiv:1810.12065 [cs, math, stat]*, October 2018b.

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A Convergence Theory for Deep Learning via Over-Parameterization. *arXiv:1811.03962 [cs, math, stat]*, November 2018c.

Sanjeev Arora, Simon S. Du, Wei Hu, Zhiyuan Li, Ruslan Salakhutdinov, and Ruosong Wang. On Exact Computation with an Infinitely Wide Neural Net. *arXiv:1904.11955 [cs, stat]*, April 2019a.

Sanjeev Arora, Simon S. Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-Grained Analysis of Optimization and Generalization for Overparameterized Two-Layer Neural Networks. January 2019b.

A. R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(3):930–945, May 1993. ISSN 0018-9448. doi: 10.1109/18.256500.

Ronen Basri, David Jacobs, Yoni Kasten, and Shira Kritchman. The Convergence Rate of Neural Networks for Learned Functions of Different Frequencies. *arXiv:1906.00425 [cs, eess, stat]*, June 2019.

Alberto Bietti and Julien Mairal. On the Inductive Bias of Neural Tangent Kernels. *arXiv:1905.12173 [cs, stat]*, May 2019.

Kenneth Blomqvist, Samuel Kaski, and Markus Heinonen. Deep convolutional Gaussian processes. *arXiv preprint arXiv:1810.03052*, 2018.

Anastasia Borovykh. A gaussian process perspective on convolutional neural networks. *arXiv preprint arXiv:1810.10798*, 2018.

John Bradshaw, Alexander G de G Matthews, and Zoubin Ghahramani. Adversarial examples, uncertainty, and transfer testing robustness in gaussian process hybrid deep networks. *arXiv preprint arXiv:1707.02476*, 2017.

Emmanuel J. Candès. Harmonic Analysis of Neural Networks. *Applied and Computational Harmonic Analysis*, 6(2):197–218, March 1999. ISSN 1063-5203. doi: 10.1006/acha.1998.0248.

Minmin Chen, Jeffrey Pennington, and Samuel Schoenholz. Dynamical Isometry and a Mean Field Theory of RNNs: Gating Enables Signal Propagation in Recurrent Neural Networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 873–882, Stockholmsmässan, Stockholm Sweden, July 2018. PMLR.

Youngmin Cho and Lawrence K. Saul. Kernel methods for deep learning. In *Advances in Neural Information Processing Systems*, pages 342–350, 2009.

Andreas Damianou and Neil Lawrence. Deep gaussian processes. In *Artificial Intelligence and Statistics*, pages 207–215, 2013.

Amit Daniely, Roy Frostig, and Yoram Singer. Toward Deeper Understanding of Neural Networks: The Power of Initialization and a Dual View on Expressivity. *arXiv:1602.05897 [cs, stat]*, February 2016.

Simon S. Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient Descent Provably Optimizes Over-parameterized Neural Networks. *arXiv:1810.02054 [cs, math, stat]*, October 2018.

Ronen Eldan and Ohad Shamir. The Power of Depth for Feedforward Neural Networks. In *Conference on Learning Theory*, pages 907–940, June 2016.

Adrià Garriga-Alonso, Laurence Aitchison, and Carl Edward Rasmussen. Deep Convolutional Networks as shallow Gaussian Processes. *arXiv:1808.05587 [cs, stat]*, August 2018.

Behrooz Ghorbani, Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Linearized two-layers neural networks in high dimension. *arXiv:1904.12191 [cs, math, stat]*, April 2019.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, May 2010. PMLR. 02641.

Tilmann Gneiting. Strictly and non-strictly positive definite functions on spheres. *Bernoulli*, 19(4): 1327–1349, September 2013. ISSN 1350-7265. doi: 10.3150/12-BEJSP06.

Boris Hanin. Which Neural Net Architectures Give Rise To Exploding and Vanishing Gradients? January 2018.

Boris Hanin and David Rolnick. How to Start Training: The Effect of Initialization and Architecture. *arXiv:1803.01719 [cs, stat]*, March 2018.

Soufiane Hayou, Arnaud Doucet, and Judith Rousseau. On the Selection of Initialization and Activation Function for Deep Neural Networks. *arXiv:1805.08266 [cs, stat]*, May 2018.

Tamir Hazan and Tommi Jaakkola. Steps Toward Deep Kernel Methods from Infinite Neural Networks. *arXiv:1508.05133 [cs]*, August 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1026–1034, 2015.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural Tangent Kernel: Convergence and Generalization in Neural Networks. *arXiv:1806.07572 [cs, math, stat]*, June 2018. 00000.

Ryo Karakida, Shotaro Akaho, and Shun-ichi Amari. Universal Statistics of Fisher Information in Deep Neural Networks: Mean Field Approach. *arXiv:1806.01316 [cond-mat, stat]*, June 2018.

Yitzhak Katznelson. *An Introduction to Harmonic Analysis*. Cambridge Mathematical Library. Cambridge University Press, Cambridge, UK ; New York, 3rd ed edition, 2004. ISBN 978-0-521-83829-0 978-0-521-54359-0.

Vinayak Kumar, Vaibhav Singh, PK Srijith, and Andreas Damianou. Deep Gaussian Processes with Convolutional Kernels. *arXiv preprint arXiv:1806.01655*, 2018.

Neil D Lawrence and Andrew J Moore. Hierarchical Gaussian process latent variable models. In *Proceedings of the 24th International Conference on Machine Learning*, pages 481–488. ACM, 2007.

Nicolas Le Roux and Yoshua Bengio. Continuous neural networks. In *Artificial Intelligence and Statistics*, pages 404–411, 2007.

Jaehoon Lee, Yasaman Bahri, Roman Novak, Sam Schoenholz, Jeffrey Pennington, and Jascha Sohl-dickstein. Deep Neural Networks as Gaussian Processes. In *International Conference on Learning Representations*, 2018.

Jaehoon Lee, Lechao Xiao, Samuel S. Schoenholz, Yasaman Bahri, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide Neural Networks of Any Depth Evolve as Linear Models Under Gradient Descent. *arXiv:1902.06720 [cs, stat]*, February 2019.

Junhong Lin and Lorenzo Rosasco. Optimal Rates for Multi-pass Stochastic Gradient Methods. page 47.

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*, April 2018.

Radford M Neal. *BAYESIAN LEARNING FOR NEURAL NETWORKS*. PhD Thesis, University of Toronto, 1995.

Roman Novak, Lechao Xiao, Jaehoon Lee, Yasaman Bahri, Daniel A Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. Bayesian Deep Convolutional Networks with Many Channels are Gaussian Processes. *arXiv preprint arXiv:1810.05148*, 2018.

Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, New York, NY, 2014. ISBN 978-1-107-03832-5.

Jeffrey Pennington and Yasaman Bahri. Geometry of Neural Network Loss Surfaces via Random Matrix Theory. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2798–2806, International Convention Centre, Sydney, Australia, August 2017. PMLR. 00006.

Jeffrey Pennington and Pratik Worah. Nonlinear random matrix theory for deep learning. In *Advances in Neural Information Processing Systems*, pages 2634–2643, 2017. 00000.

Jeffrey Pennington and Pratik Worah. The Spectrum of the Fisher Information Matrix of a Single-Hidden-Layer Neural Network. In *Advances in Neural Information Processing Systems 31*, page 10, 2018.

Jeffrey Pennington, Samuel Schoenholz, and Surya Ganguli. Resurrecting the sigmoid in deep learning through dynamical isometry: Theory and practice. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4788–4798. Curran Associates, Inc., 2017a. 00004.

Jeffrey Pennington, Samuel S. Schoenholz, and Surya Ganguli. Resurrecting the sigmoid in deep learning through dynamical isometry: Theory and practice. *arXiv:1711.04735 [cs, stat]*, November 2017b. 00005.

Jeffrey Pennington, Samuel S. Schoenholz, and Surya Ganguli. The Emergence of Spectral Universality in Deep Networks. *arXiv:1802.09979 [cs, stat]*, February 2018.

George Philipp and Jaime G. Carbonell. The Nonlinearity Coefficient - Predicting Overfitting in Deep Neural Networks. *arXiv:1806.00179 [cs, stat]*, May 2018. 00000.

Ben Poole, Subhaneil Lahiri, Maithreyi Raghu, Jascha Sohl-Dickstein, and Surya Ganguli. Exponential expressivity in deep neural networks through transient chaos. In *Advances In Neural Information Processing Systems*, pages 3360–3368, 2016. 00047.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred A. Hamprecht, Yoshua Bengio, and Aaron Courville. On the Spectral Bias of Neural Networks. *arXiv:1806.08734 [cs, stat]*, June 2018.

Garvesh Raskutti, Martin J. Wainwright, and Bin Yu. Early stopping and non-parametric regression: An optimal data-dependent stopping rule. *arXiv:1306.3574 [stat]*, June 2013.

I. J. Schoenberg. Positive definite functions on spheres. *Duke Mathematical Journal*, 9(1):96–108, March 1942. ISSN 0012-7094, 1547-7398. doi: 10.1215/S0012-7094-42-00908-6.

Samuel S. Schoenholz, Justin Gilmer, Surya Ganguli, and Jascha Sohl-Dickstein. Deep Information Propagation. 2017.

Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, Mass, 2002. ISBN 978-0-262-19475-4.

Alex J. Smola, Zoltán L. Óvári, and Robert C Williamson. Regularization with Dot-Product Kernels. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 308–314. MIT Press, 2001.

Sho Sonoda and Noboru Murata. Neural Network with Unbounded Activation Functions is Universal Approximator. *Applied and Computational Harmonic Analysis*, 43(2):233–268, September 2017. ISSN 10635203. doi: 10.1016/j.acha.2015.12.005.

P.K. Suetin. Ultraspherical polynomials - Encyclopedia of Mathematics. https://www.encyclopediaofmath.org/index.php/Ultraspherical_polynomials.

Guillermo Valle-Pérez, Chico Q. Camargo, and Ard A. Louis. Deep learning generalizes because the parameter-function map is biased towards simple functions. *arXiv:1805.08522 [cs, stat]*, May 2018.

Mark van der Wilk, Carl Edward Rasmussen, and James Hensman. Convolutional Gaussian Processes. In *Advances in Neural Information Processing Systems 30*, pages 2849–2858, 2017.

Yuting Wei, Fanny Yang, and Martin J Wainwright. Early stopping for kernel boosting algorithms: A general analysis with localized complexities. page 11.

Christopher K I Williams. Computing with Infinite Networks. In *Advances in Neural Information Processing Systems*, page 7, 1997.

Andrew G Wilson, Zhiting Hu, Ruslan R Salakhutdinov, and Eric P Xing. Stochastic Variational Deep Kernel Learning. In *Advances in Neural Information Processing Systems*, pages 2586–2594, 2016a.

Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P Xing. Deep kernel learning. In *Artificial Intelligence and Statistics*, pages 370–378, 2016b.

Lechao Xiao, Yasaman Bahri, Sam Schoenholz, and Jeffrey Pennington. Training ultra-deep CNNs with critical initialization. In *NIPS Workshop*, 2017.

Bo Xie, Yingyu Liang, and Le Song. Diverse Neural Network Learns True Target Functions. *arXiv:1611.03131 [cs, stat]*, November 2016.

Yuan Xu and E. W. Cheney. Strictly Positive Definite Functions on Spheres. *Proceedings of the American Mathematical Society*, 116(4):977–981, 1992. ISSN 0002-9939. doi: 10.2307/2159477.

Zhi-Qin John Xu, Yaoyu Zhang, and Yanyang Xiao. Training behavior of deep neural network in frequency domain. *arXiv:1807.01251 [cs, math, stat]*, July 2018.

Zhi-Qin John Xu, Yaoyu Zhang, Tao Luo, Yanyang Xiao, and Zheng Ma. Frequency Principle: Fourier Analysis Sheds Light on Deep Neural Networks. *arXiv:1901.06523 [cs, stat]*, January 2019.

Zhiqin John Xu. Understanding training and generalization in deep learning by Fourier analysis. *arXiv:1808.04295 [cs, math, stat]*, August 2018.

Greg Yang. Scaling Limits of Wide Neural Networks with Weight Sharing: Gaussian Process Behavior, Gradient Independence, and Neural Tangent Kernel Derivation. *arXiv:1902.04760 [cond-mat, physics:math-ph, stat]*, February 2019.

Greg Yang and Sam S. Schoenholz. Deep Mean Field Theory: Layerwise Variance and Width Variation as Methods to Control Gradient Explosion. February 2018.

Greg Yang and Samuel S. Schoenholz. Mean Field Residual Network: On the Edge of Chaos. In *Advances in Neural Information Processing Systems*, 2017.

Greg Yang, Jeffrey Pennington, Vinay Rao, Jascha Sohl-Dickstein, and Samuel S. Schoenholz. A Mean Field Theory of Batch Normalization. *arXiv:1902.08129 [cond-mat]*, February 2019.

Yuan Yao, Lorenzo Rosasco, and Andrea Caponnetto. On Early Stopping in Gradient Descent Learning. *Constructive Approximation*, 26(2):289–315, August 2007. ISSN 0176-4276, 1432-0940. doi: 10.1007/s00365-006-0663-2.

Yaoyu Zhang, Zhi-Qin John Xu, Tao Luo, and Zheng Ma. Explicitizing an Implicit Bias of the Frequency Principle in Two-layer Neural Networks. *arXiv:1905.10264 [cs, stat]*, May 2019.

Difan Zou, Yuan Cao, Dongruo Zhou, and Quanquan Gu. Stochastic Gradient Descent Optimizes Over-parameterized Deep ReLU Networks. *arXiv:1811.08888 [cs, math, stat]*, November 2018.

A RELATED WORKS

The Gaussian process behavior of neural networks was found by Neal (1995) for shallow networks and then extended over the years to different settings and architectures (Williams, 1997; Le Roux and Bengio, 2007; Hazan and Jaakkola, 2015; Daniely et al., 2016; Lee et al., 2018; Matthews et al., 2018; Novak et al., 2018). This connection was exploited implicitly or explicitly to build new models (Cho and Saul, 2009; Lawrence and Moore, 2007; Damianou and Lawrence, 2013; Wilson et al., 2016a;b; Bradshaw et al., 2017; van der Wilk et al., 2017; Kumar et al., 2018; Blomqvist et al., 2018; Borovykh, 2018; Garriga-Alonso et al., 2018; Novak et al., 2018; Lee et al., 2018). The Neural Tangent Kernel is a much more recent discovery by Jacot et al. (2018) and later Allen-Zhu et al. (2018a;c;b); Du et al. (2018); Arora et al. (2019b); Zou et al. (2018) came upon the same reasoning independently. Like CK, NTK has also been applied toward building new models or algorithms (Arora et al., 2019a; Achiam et al., 2019).

Closely related to the discussion of CK and NTK is the signal propagation literature, which tries to understand how to prevent pathological behaviors in randomly initialized neural networks when they are deep (Poole et al., 2016; Schoenholz et al., 2017; Yang and Schoenholz, 2017; 2018; Hanin, 2018; Hanin and Rolnick, 2018; Chen et al., 2018; Yang et al., 2019; Pennington et al., 2017a; Hayou et al., 2018; Philipp and Carbonell, 2018). This line of work can trace its original at least to the advent of the *Glorot* and *He* initialization schemes for deep networks (Glorot and Bengio, 2010; He et al., 2015). The investigation of *forward signal propagation*, or how random neural networks change with depth, corresponds to studying the infinite-depth limit of CK, and the investigation of *backward signal propagation*, or how gradients of random networks change with depth, corresponds to studying the infinite-depth limit of NTK. Some of the quite remarkable results from this literature includes how to train a 10,000 layer CNN (Xiao et al., 2017) and that, counterintuitively, batch normalization causes gradient explosion (Yang et al., 2019).

This signal propagation perspective can be refined via random matrix theory (Pennington et al., 2017b; 2018). In these works, free probability is leveraged to compute the singular value distribution of the input-output map given by the random neural network, as the input dimension and width tend to infinity together. Other works also investigate various questions of neural network training and generalization from the random matrix perspective (Pennington and Worah, 2017; Pennington and Bahri, 2017; Pennington and Worah, 2018).

Yang (2019) presents a common framework, known as *Tensor Programs*, unifying the GP, NTK, signal propagation, and random matrix perspectives, as well as extending them to new scenarios, like recurrent neural networks. It proves the existence of and allows the computation of a large number of infinite-width limits (including ones relevant to the above perspectives) by expressing the quantity of interest as the output of a computation graph and then manipulating the graph mechanically.

Several other works also adopt a spectral perspective on neural networks (Candès, 1999; Sonoda and Murata, 2017; Eldan and Shamir, 2016; Barron, 1993; Xu et al., 2018; Zhang et al., 2019; Xu et al., 2019; Xu, 2018); here we highlight a few most relevant to us. Rahaman et al. (2018) studies the *real Fourier frequencies* of relu networks and perform experiments on real data as well as synthetic ones. They convincingly show that relu networks learn low frequencies components first. They also investigate the subtleties when the data manifold is low dimensional and embedded in various ways in the input space. In contrast, our work focuses on the spectra of the CK and NTK (which indirectly informs the Fourier frequencies of a typical network). Nevertheless, our results are complementary to theirs, as they readily explain the low frequency bias in relu that they found. Karakida et al. (2018) studies the spectrum of the Fisher information matrix, which share the nonzero eigenvalues with the NTK. They compute the mean, variance, and maximum of the eigenvalues Fisher eigenvalues (taking the width to infinity first, and then considering finite amount of data sampled iid from a Gaussian). In comparison, our spectral results yield all eigenvalues of the NTK (and thus also all nonzero eigenvalues of the Fisher) as well as eigenfunctions.

Finally, we note that several recent works (Xie et al., 2016; Bietti and Mairal, 2019; Basri et al., 2019; Ghorbani et al., 2019) studied one-hidden layer neural networks over the sphere, building on Smola et al. (2001)'s observation that spherical harmonics diagonalize dot product kernels, with the latter two concurrent to us. This is in contrast to the focus on boolean cube here, which allows us to study the fine-grained effect of hyperparameters on the spectra, leading to a variety of insights into neural networks' generalization properties.

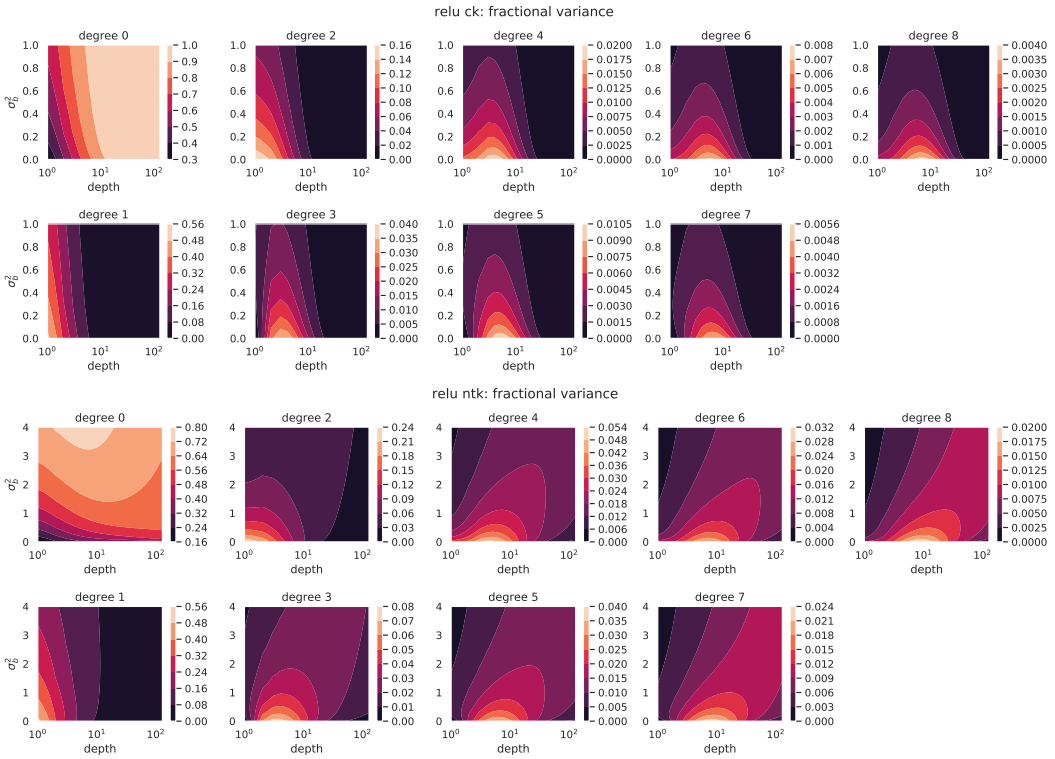


Figure 5: **2D contour plots of how fractional variance of each degree varies with σ_b^2 and depth, fixing $\sigma_w^2 = 2$, for relu CK and NTK**. For each degree k, and for each selected fractional variance value, we plot the level curve of $(\text{depth}, \sigma_b^2)$ achieving this value. The color indicates the fractional variance, as given in the color bars.

B Visualizing the Spectral Effects of σ_w^2, σ_b^2, and Depth

While in the main text, we summarized several trends of interest kn several plots, they do not give the entire picture of how eigenvalues and fractional variances vary with σ_w^2, σ_b^2, and depth. Here we try to present this relationship more completely in a series of contour plots. Fig. 5 shows how varying depth and σ_b^2 changes the fractional variances of each degree, for relu CK and NTK. We are fixing $\sigma_w^2 = 2$ in the CK plots, as the fractional variances only depend on the ratio σ_b^2/σ_w^2; even though this is not true for relu NTK, we fix $\sigma_w^2 = 2$ as well for consistency. For erf, however, the fractional variance will crucially depend on both σ_w^2 and σ_b^2, so we present 3D contour plots of how σ_w^2, σ_b^2, and depth changes fractional variance in Fig. 8. Complementarily, we also show in Figs. 6 and 7 a few slices of these 3D contour plots for different fixed values of σ_b^2, for erf CK and NTK.

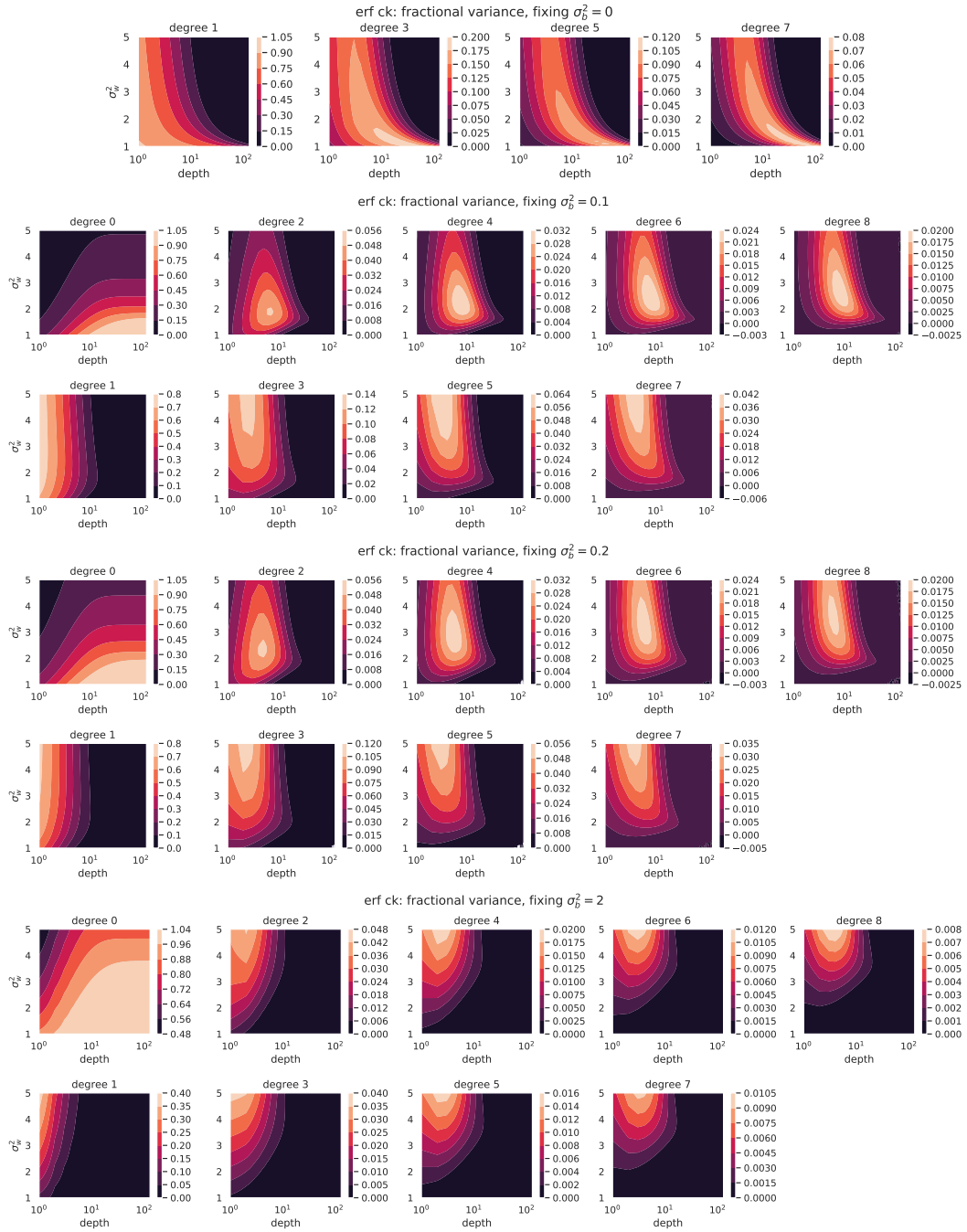


Figure 6: **2D contour plots of how fractional variance of each degree varies with σ_w^2 and depth, for different slices of σ_b^2, for erf NTK**. These plots essentially show slices of the NTK 3D contour plots in Fig. 8. For $\sigma_b = 0$, μ_k for all even degrees k are 0, so we omit the plots. Note the rapid change in the shape of the contours for odd degrees, going from $\sigma_b^2 = 0$ to $\sigma_b^2 = 0.1$. This is reflected in Fig. 8 as well.

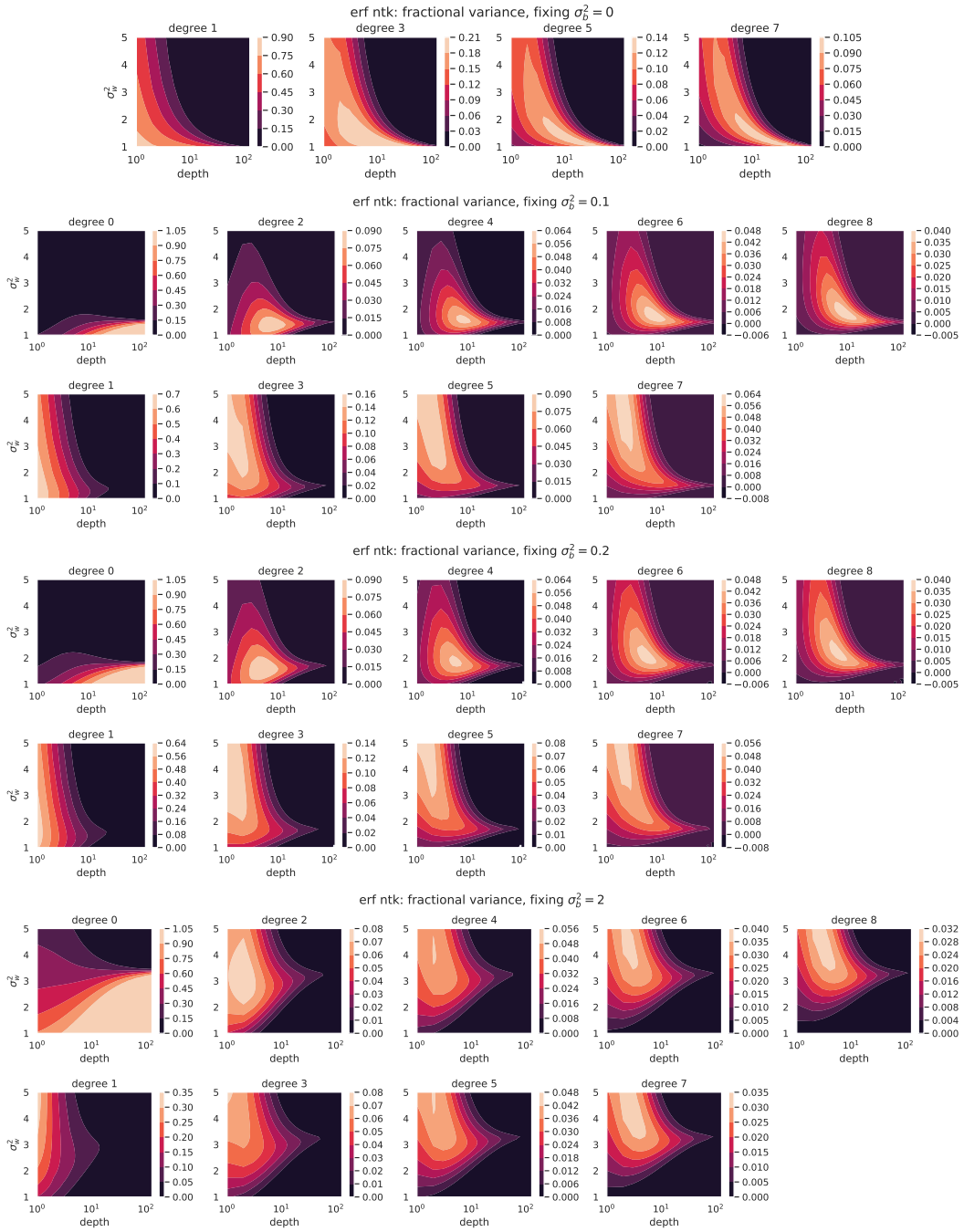


Figure 7: **2D contour plots of how fractional variance of each degree varies with σ_w^2 and depth, for different slices of σ_b^2, for erf CK**. These plots essentially show slices of the CK 3D contour plots in Fig. 8. For $\sigma_b = 0$, μ_k for all even degrees k are 0, so we omit the plots. Note the rapid change in the shape of the contours for odd degrees, going from $\sigma_b^2 = 0$ to $\sigma_b^2 = 0.1$. This is reflected in Fig. 8 as well.

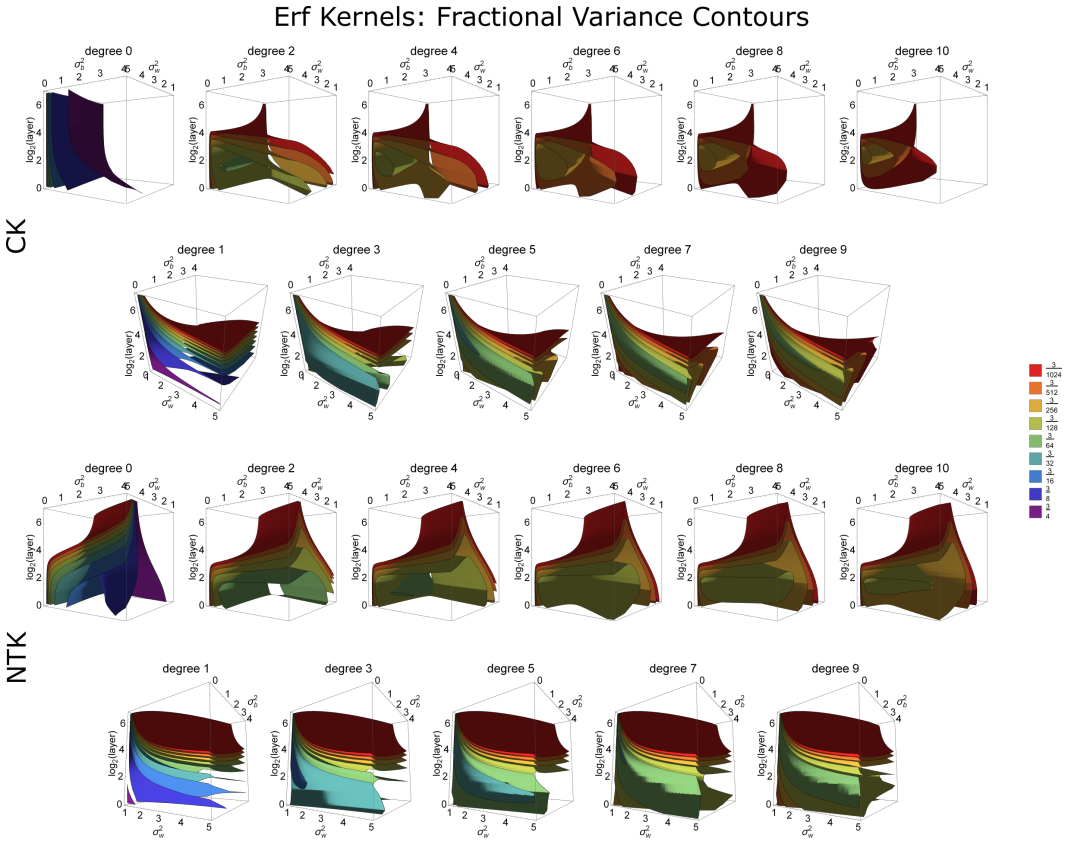


Figure 8: **3D contour plots of how fractional variance of each degree varies with σ_w^2, σ_b^2 and $\log_2(\text{depth})$, for erf CK and NTK**. For each value of fractional variance, as given in the legend on the right, we plot the level surface in the $(\sigma_w^2, \sigma_b^2, \log_2(\text{depth}))$-space achieving this value in the corresponding color. The closer to blue the color, the higher the value. Note that the contour for the highest values in higher degree plots "floats in mid-air", implying that there is an optimal depth for learning features of that degree that is not particularly small nor particularly big.

C EXPERIMENTAL DETAILS

Fig. 3(a), (b) and (c) differ in the set of hyperparameters they involve (to be specified below), but in all of them, we train relu networks against a randomly generated ground truth multilinear polynomial, with input space $\square^{128}$ and L2 loss $L(f) = \mathbb{E}_{x \in \square^d}(f(x) - f^*(x))^2$.

Training We perform SGD with batch size 1000. In each iteration, we freshly sample a new batch, and we train for a total of 100,000 iterations, so the network potentially sees 10^8 different examples. At every 1000 iterations, we validate the current network on a freshly drawn batch of 10,000 examples. We thus record a total of 100 validation losses, and we take the lowest to be the "best validation loss."

Generating the Ground Truth Function The ground truth function $f^*(x)$ is generated by first sampling 10 monomials $m_1, \ldots, m_{10}$ of degree k, then randomly sampling 10 coefficients $a_1, \ldots, a_{10}$ for them. The final function is obtained by normalizing $\{a_i\}$ such that the sum of their squares is 1:

$$f^*(x) \stackrel{\text{def}}{=} \sum_{i=1}^{10} a_i m_i / \sum_{j=1}^{10} a_j^2.$$

Hyperparameters for Fig. 3(a)

- The learning rate is half the theoretical maximum learning rate[6] $\frac{1}{2}\max(\mu_0, \mu_1)^{-1}$
- Ground truth degree $k \in \{0, 1, 2, 3\}$
- Depth $\in \{0, \ldots, 10\}$
- activation = relu
- $\sigma_w^2 = 2$
- $\sigma_b^2 = 0$
- width = 1000
- 10 random seeds per hyperparameter combination
- training last layer (marked "ck"), or all layers (marked "ntk")

Hyperparameters for Fig. 3(b)

- The learning rate is half the theoretical maximum learning rate $\frac{1}{2}\max(\mu_0, \mu_1)^{-1}$
- Ground truth degree $k \in \{0, 1, 2, 3\}$
- Depth $\in \{0, \ldots, 10\}$
- activation = relu
- $\sigma_w^2 = 2$
- $\sigma_b^2 = 0$
- width = 1000
- 100 random seeds per hyperparameter combination
- training last layer weight and bias only

Hyperparameters for Fig. 3(c)

- The learning rate $\in \{0.05, 0.1, 0.5\}$
- Ground truth degree $k \in \{0, 1, \ldots, 6\}$
- Depth $\in \{1, \ldots, 5\}$
- activation $\in \{\mathrm{relu}, \mathrm{erf}\}$

[6]Note that, because the L2 loss here is $L(f) = \mathbb{E}_{x \in \square^d}(f(x) - f^*(x))^2$, the maximum learning rate is $\lambda_{max}^{-1} = \max(\mu_0, \mu_1)^{-1}$ (see Thm 4.1). If we instead adopt the convention $L(f) = \mathbb{E}_{x \in \square^d} \frac{1}{2}(f(x) - f^*(x))^2$, then the maximum learning rate would be $2\lambda_{max}^{-1} = 2\max(\mu_0, \mu_1)^{-1}$

- $\sigma_w^2 = 2$ for relu, but $\sigma_w^2 \in \{1, 2, \ldots, 5\}$ for erf
- $\sigma_b^2 \in \{0, 1, \ldots, 4\}$
- width = 1000
- 1 random seed per hyperparameter combination
- Training all layers

D REVIEW OF THE THEORY OF NEURAL TANGENT KERNELS

D.1 CONVERGENCE OF INFINITE-WIDTH KERNELS AT INITIALIZATION

Conjugate Kernel Via a central-limit-like intuition, each unit $h^l(x)_\alpha$ of Eq. (MLP) should behave like a Gaussian as width $n^{l-1} \to \infty$, as it is a sum of a large number of roughly independent random variables (Poole et al., 2016; Schoenholz et al., 2017; Yang and Schoenholz, 2017). The devil, of course, is in what "roughly independent" means and how to apply the central limit theorem (CLT) to this setting. It can be done, however, (Lee et al., 2018; Matthews et al., 2018; Novak et al., 2018), and in the most general case, using a "Gaussian conditioning" technique, this result can be rigorously generalized to almost any architecture Yang (2019). In any case, the consequence is that, for any finite set $S \subseteq \mathcal{X}$,

$$\{h_\alpha^l(x)\}_{x \in S} \quad \text{converges in distribution to} \quad \mathcal{N}(0, \Sigma^l(S, S)),$$

as $\min\{n^1, \ldots, n^{l-1}\} \to \infty$, where Σ^l is the CK as given in Eq. (CK).

Neural Tangent Kernel By a slightly more involved version of the "Gaussian conditioning" technique, Yang (2019) also showed that, for any $x, y \in \mathcal{X}$,

$$\langle \nabla_\theta h^L(x), \nabla_\theta h^L(y) \rangle \quad \text{converges almost surely to} \quad \Theta^L(x, y)$$

as the widths tend to infinity, where Θ^l is the NTK as given in Eq. (NTK).

D.2 FAST EVALUATIONS OF CK AND NTK

For certain ϕ like relu or erf, V_ϕ and V'_ϕ can be evaluated very quickly, so that both the CK and NTK can be computed in $O(|\mathcal{X}|^2 L)$ time, where $\mathcal{X}$ is the set of points we want to compute the kernel function over, and L is the number of layers.

Fact D.1 (Cho and Saul (2009))**.** *For any kernel K*

$$\begin{aligned}
\mathrm{V}_{\mathrm{relu}}(K)(x, x') &= \frac{1}{2\pi}(\sqrt{1 - c^2} + (\pi - \arccos c)c)\sqrt{K(x, x)K(x', x')} \\
\mathrm{V}'_{\mathrm{relu}}(K)(x, x') &= \frac{1}{2\pi}(\pi - \arccos c)
\end{aligned}$$

where $c = K(x, x')/\sqrt{K(x, x)K(x', x')}$.

Fact D.2 (Neal (1995))**.** *For any kernel K,*

$$\begin{aligned}
\mathrm{V}_{\mathrm{erf}}(K)(x, x') &= \frac{2}{\pi} \arcsin \frac{K(x, x')}{\sqrt{(K(x, x) + 0.5)(K(x', x') + 0.5)}} \\
\mathrm{V}'_{\mathrm{erf}}(K)(x, x') &= \frac{4}{\pi\sqrt{(1 + 2K(x, x))(1 + 2K(x', x')) - 4K(x, x')^2}}.
\end{aligned}$$

Fact D.3. *Let $\phi(x) = \exp(x/\sigma)$ for some $\sigma > 0$. For any kernel K,*

$$\mathrm{V}_\phi(K)(x, x') = \exp\left(\frac{K(x, x) + 2K(x, x') + K(x', x')}{2\sigma^2}\right).$$

D.3 LINEAR EVOLUTION OF NEURAL NETWORK UNDER GD

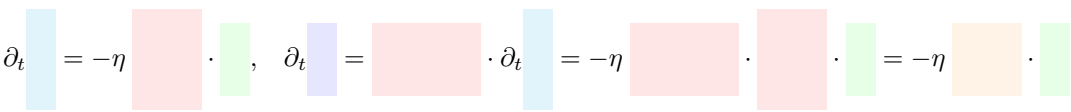
Remarkably, the NTK governs the evolution of the neural network function under gradient descent in the infinite-width limit. First, let's consider how the parameters θ and the neural network function f evolve under continuous time gradient flow. Suppose f is only defined on a finite input space $\mathcal{X} = \{x^1, \ldots, x^k\}$. We will visualize

$$f(\mathcal{X}) = \begin{bmatrix} f(x^1) \\ \vdots \\ f(x^k) \end{bmatrix}, \quad \nabla_f \mathcal{L} = \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial f(x^1)} \\ \vdots \\ \frac{\partial \mathcal{L}}{\partial f(x^k)} \end{bmatrix}, \quad \theta = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_n \end{bmatrix}, \quad \nabla_\theta f = \begin{bmatrix} \frac{\partial f(x^1)}{\partial \theta_1} & \cdots & \frac{\partial f(x^k)}{\partial \theta_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial f(x^1)}{\partial \theta_n} & \cdots & \frac{\partial f(x^k)}{\partial \theta_n} \end{bmatrix}$$

(best viewed in color). Then under continuous time gradient descent with learning rate η,

$$\begin{aligned} \partial_t \theta_t &= -\eta \nabla_\theta \mathcal{L}(f_t) = -\eta \nabla_\theta f_t \cdot \nabla_f \mathcal{L}(f_t), \\ \partial_t f_t &= \nabla_\theta f_t^\top \cdot \partial_t \theta_t = -\eta \nabla_\theta f_t^\top \cdot \nabla_\theta f_t \cdot \nabla_f \mathcal{L}(f_t) = -\eta \Theta_t \cdot \nabla_f \mathcal{L}(f_t) \end{aligned} \tag{10}$$

where $\Theta_t = \nabla_\theta f_t^\top \cdot \nabla_\theta f_t \in \mathbb{R}^{k \times k}$ is of course the (finite width) NTK. These equations can be visualized as



Thus f undergoes kernel gradient descent with (functional) loss $\mathcal{L}(f)$ and kernel Θ_t. This kernel Θ_t of course changes as f evolves, but remarkably, it in fact stays constant for f being an infinitely wide MLP (Jacot et al., 2018):

$$\partial_t f_t = -\eta \Theta \cdot \nabla_f \mathcal{L}(f_t), \tag{Training All Layers}$$

where Θ is the infinite-width NTK corresponding to f. A similar equation holds for the CK Σ if we train only the last layer,

$$\partial_t f_t = -\eta \Sigma \cdot \nabla_f \mathcal{L}(f_t). \tag{Training Last Layer}$$

If $\mathcal{L}$ is the square loss against a ground truth function f^*, then $\nabla_f \mathcal{L}(f_t) = \frac{1}{2k} \nabla_f \|f_t - f^*\|^2 = \frac{1}{k}(f_t - f^*)$, and the equations above become linear differential equations. However, typically we only have a training set $\mathcal{X}^{\text{train}} \subseteq \mathcal{X}$ of size far less than $|\mathcal{X}|$. In this case, the loss function is effectively

$$\mathcal{L}(f) = \frac{1}{2|\mathcal{X}^{\text{train}}|} \sum_{x \in \mathcal{X}^{\text{train}}} (f(x) - f^*(x))^2,$$

with functional gradient

$$\nabla_f \mathcal{L}(f) = \frac{1}{|\mathcal{X}^{\text{train}}|} D^{\text{train}} \cdot (f - f^*),$$

where D^{train} is a diagonal matrix of size $k \times k$ whose diagonal is 1 on $x \in \mathcal{X}^{\text{train}}$ and 0 else. Then our function still evolves linearly

$$\partial_t f_t = -\eta (K \cdot D^{\text{train}}) \cdot (f_t - f^*) \tag{11}$$

where K is the CK or the NTK depending on which parameters are trained.

D.4 RELATIONSHIP TO GAUSSIAN PROCESS INFERENCE.

Recall that the initial f_0 in Eq. (11) is distributed as a Gaussian process $\mathcal{N}(0, \Sigma)$ in the infinite width limit. As Eq. (11) is a linear differential equation, the distribution of f_t will remain a Gaussian process for all t, whether K is CK or NTK. Under suitable conditions, it can be shown that (Lee et al., 2019), in the limit as $t \to \infty$, if we train only the last layer, then the resulting function f_∞ is distributed as a Gaussian process with mean $\bar{f}_\infty$ given by

$$\bar{f}_\infty(x) = \Sigma(x, \mathcal{X}^{\text{train}}) \Sigma(\mathcal{X}^{\text{train}}, \mathcal{X}^{\text{train}})^{-1} f^*(\mathcal{X}^{\text{train}})$$

and kernel $\mathrm{Var}\, f_\infty$ given by

$$\mathrm{Var}\, f_\infty(x, x') = \Sigma(x, x') - \Sigma(x, \mathcal{X}^{\mathrm{train}})\Sigma(\mathcal{X}^{\mathrm{train}}, \mathcal{X}^{\mathrm{train}})^{-1}\Sigma(\mathcal{X}^{\mathrm{train}}, x').$$

These formulas precisely described the posterior distribution of f given prior $\mathcal{N}(0, \Sigma)$ and data $\{(x, f^*(x))\}_{x\in\mathcal{X}^{\mathrm{train}}}$.

If we train all layers, then similarly as $t \to \infty$, the function f_∞ is distributed as a Gaussian process with mean $\bar{f}_\infty$ given by (Lee et al., 2019)

$$\bar{f}_\infty(x) = \Theta(x, \mathcal{X}^{\mathrm{train}})\Theta(\mathcal{X}^{\mathrm{train}}, \mathcal{X}^{\mathrm{train}})^{-1} f^*(\mathcal{X}^{\mathrm{train}}).$$

This is, again, the mean of the Gaussian process posterior given prior $\mathcal{N}(0, \Theta)$ and the training data $\{(x, f^*(x))\}_{x\in\mathcal{X}^{\mathrm{train}}}$. However, the kernel of f_∞ is no longer the kernel of this posterior, but rather is an expression involving both the NTK Θ and the CK Σ; see Lee et al. (2019).

In any case, we can make the following informal statement in the limit of large width

> *Training the last layer (resp. all layers) of an MLP infinitely long, in expectation, yields the mean prediction of the GP inference given prior $\mathcal{N}(0, \Sigma)$ (resp. $\mathcal{N}(0, \Theta)$).*

Discrete Time GD In regards to training, in the limit of large width and infinite data, the function g represented by our neural network evolves like

$$g^{t+1} = g^t - \alpha K(g^t - g^*), \quad t = 0, 1, 2, \ldots,$$

when trained under full batch GD (with the entire population) with L2 loss, ground truth g^*, and learning rate α, starting from randomly initialization Jacot et al. (2018). If we train only the last layer, then K is the CK; if we train all layers, then K is the NTK. Given an eigendecomposition of K as in Eq. (1), if $g^0 - g^* = \sum_i a_i f_i$ is the decomposition of g^0 in the basis $\{f_i\}$, then one can easily deduce that

$$g^t - g^* = \sum_i a_i (1 - \alpha\lambda_i)^t f_i.$$

If λ_i is large, then $a_i(1 - \alpha\lambda_i)^t$ drops to zero very quickly, i.e. "g^t learns the ith component of g^* rapidly", provided α is not too large. Conversely, if λ_i is small, then the corresponding component of g^* is learned very slowly. If the coefficient of g^* w.r.t f_i is large, then this could mean g^* takes a very long time to learn. From this discrete time perspective, one can also make the prediction that α must be at most $2\,(\max_i \lambda_i)^{-1}$ in order for gradient descent to converge.

E A Brief Review of Hilbert-Schmidt Operators and Their Spectral Theory

In this section, we briefly review the theory of Hilbert-Schmidt kernels, and more importantly, to properly define the notion of eigenvalues and eigenfunctions. A function $K : \mathcal{X}^2 \to \mathbb{R}$ is called a *Hilbert-Schmidt* operator if $K \in L^2(\mathcal{X} \times \mathcal{X})$, i.e.

$$\|K\|_{HS}^2 \stackrel{\text{def}}{=} \mathop{\mathbb{E}}_{x,y\sim\mathcal{X}} K(x, y)^2 < \infty.$$

$\|K\|_{HS}^2$ is known as the *Hilbert-Schmidt* norm of K. K is called *symmetric* if $K(x, y) = K(y, x)$ and *positive definite (resp. semidefinite)* if

$$\mathop{\mathbb{E}}_{x,y\sim\mathcal{X}} f(x)K(x, y)f(y) > 0 \text{ (resp. } \geq 0) \quad \text{for all } f \in L^2(\mathcal{X}) \text{ not a.e. zero.}$$

A spectral theorem (Mercer's theorem) holds for Hilbert-Schmidt operators.

Fact E.1. *If K is a symmetric positive semidefinite Hilbert-Schmidt kernel, then there is a sequence of scalars $\lambda_i \geq 0$ (eigenvalues) and functions $f_i \in L^2(\mathcal{X})$ (eigenfunctions), for $i \in \mathbb{N}$, such that*

$$\forall i, j,\ \langle f_i, f_j\rangle = \mathbb{I}(i = j), \quad \textit{and} \quad K(x, y) = \sum_{i\in\mathbb{N}} \lambda_i f_i(x) f_i(y)$$

where the convergence is in $L^2(\mathcal{X} \times \mathcal{X})$ norm.

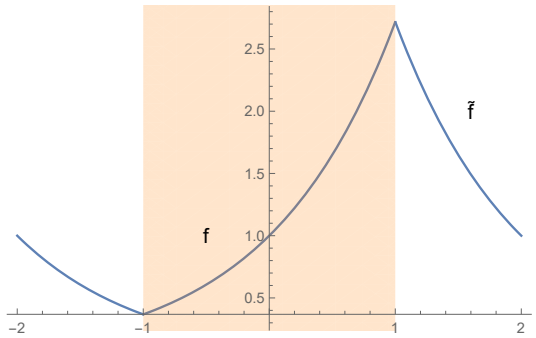


Figure 9: Example of $\tilde{f}$ given f

This theorem allows us to speak of the eigenfunctions and eigenvalues, which are important for training and generalization considerations when K is a kernel used in machine learning, as discussed in the main text.

A sufficient condition for K to be a Hilbert-Schmidt kernel in our case (concerning only probability measure on $\mathcal{X}$) is just that K is bounded. All Ks in this paper satisfy this property.

F Eigendecomposition of Neural Kernel on Different Domains

F.1 Boolean Cube

From the Fourier Series Perspective. We continue from the discussion of the boolean cube in the main text. Recall that $\mathcal{T}_\Delta$ is the shift operator on functions that sends $\Phi(\cdot)$ to $\Phi(\cdot - \Delta)$. Notice that, if we let $\Phi(t) = e^{\kappa t}$ for some $\kappa \in \mathbb{C}$, then $\mathcal{T}_\Delta \Phi(s) = e^{-\kappa\Delta} \cdot e^{\kappa t}$. Thus Φ is an "eigenfunction" of the operator $\mathcal{T}_\Delta$ with eigenvalue $e^{-\kappa\Delta}$. In particular, this implies that

Proposition F.1. *Suppose $\Phi(t) = e^{t/\sigma^2}$, as in the case when K is the CK or NTK of a 1-layer neural network with nonlinearity $\exp(\cdot/\sigma)$, up to multiplicative constant (Fact D.3). Then the eigenvalue μ_k over the boolean cube $\square^d$ equals*

$$\mu_k = 2^{-d}(1 - \exp(-\Delta/\sigma^2))^k (1 + \exp(-\Delta/\sigma^2))^{d-k} \cdot \exp(1/\sigma^2)$$

where $\Delta = 2/d$.

It would be nice if we can express any Φ as a linear combination of exponentials, so that Eq. (5) simplifies in the fashion of Prop F.1 — this is precisely the idea of Fourier series.

We will use the theory of Fourier analysis on the circle, and for this we need to discuss periodic functions. Let $\tilde{\Phi} : [-2, 2] \to \mathbb{R}$ be defined as

$$\tilde{\Phi}(x) = \begin{cases} \Phi(x) & \text{if } x \in [-1, 1] \\ \Phi(2 - x) & \text{if } x \in [1, 2] \\ \Phi(-2 - x) & \text{if } x \in [-2, -1]. \end{cases}$$

See Fig. 9 for an example illustration. Note that if Φ is continuous on $[-1, 1]$, then $\tilde{\Phi}$ is continuous as a periodic function on $[-2, 2]$.

The Fourier basis on functions over $[-2, 2]$ is the collection $\{t \mapsto e^{\frac{1}{2}\pi i s t}\}_{s \in \mathbb{Z}}$. Under generic conditions (for example if $\Psi \in L^2[-2, 2]$), a function Ψ has an associated Fourier series $\sum_{s \in \mathbb{Z}} \hat{\Psi}(s) e^{\frac{1}{2}\pi i s t}$. We briefly review basic facts of Fourier analysis on the circle. Recall the following notion of *functions of bounded variation*.

Definition F.2. A function $f : [a, b] \to \mathbb{R}$ is said to have *bounded variation* if

$$\sup_P \sum_{i=0}^{n_P - 1} |f(x_{i+1}) - f(x_i)| < \infty,$$

where the supremum is taken over all partitions P of the interval $[a, b]$,

$$P = \{x_0, \ldots, x_{n_P}\}, \quad x_0 \le x_1 \le \cdots \le x_{n_P}.$$

Intuitively, a function of bounded variation has a graph (in $[a, b] \times \mathbb{R}$) of finite length.

Fact F.3 (Katznelson (2004))**.** *A bounded variation function $f : [-2, 2] \to \mathbb{R}$ that is periodic (i.e. $f(-2) = f(2)$) has a pointwise-convergent Fourier series:*

$$\lim_{T\to\infty} \sum_{s\in[-T,T]} \hat{\Psi}(s) e^{\frac{1}{2}\pi i s t} \to \Psi(t),\ \forall t \in [-2, 2].$$

From this fact easily follows the following lemma.

Lemma F.4. *Suppose Φ is continuous and has bounded variation on $[-1, 1]$. Then $\tilde{\Phi}$ is also continuous and has bounded variation, and its Fourier Series (on $[-2, 2]$) converges pointwise to $\tilde{\Phi}$.*

Proof. $\tilde{\Phi}$ is obviously continuous and has bounded variation as well, and from Fact F.3, we know a periodic continuous function with bounded variation has a pointwise-convergent Fourier Series. □

Certainly, $\mathcal{T}_\Delta$ sends continuous bounded variation functions to continuous bounded variation functions. Because $\mathcal{T}_\Delta e^{\frac{1}{2}\pi i s t} = e^{-\frac{1}{2}\pi i s \Delta} e^{\frac{1}{2}\pi i s t}$,

$$\mathcal{T}_\Delta \sum_{s\in\mathbb{Z}} \hat{\Psi}(s) e^{\frac{1}{2}\pi i s t} = \sum_{s\in\mathbb{Z}} \hat{\Psi}(s) e^{-\frac{1}{2}\pi i s \Delta} e^{\frac{1}{2}\pi i s t}$$

whenever both sides are well defined. If Ψ is continuous and has bounded variation then $\mathcal{T}_\Delta \Psi$ is also continuous and has bounded variation, and thus its Fourier series, the RHS above, converges pointwise to $\mathcal{T}_\Delta \Psi$.

Now, observe

$$\begin{aligned}
(I - \mathcal{T}_\Delta)^k (I + \mathcal{T}_\Delta)^{d-k} \tilde{\Phi}(x) &= \sum_{r=0}^{d} C_r^{d-k,k} \tilde{\Phi}\left(x - r\Delta\right) \\
(I - \mathcal{T}_\Delta)^k (I + \mathcal{T}_\Delta)^{d-k} \tilde{\Phi}(1) &= \sum_{r=0}^{d} C_r^{d-k,k} \Phi\left(\left(\frac{d}{2} - r\right)\Delta\right) \\
&= \mu_k
\end{aligned}$$

Expressing the LHS in Fourier basis, we obtain

Theorem F.5.

$$\mu_k = \sum_{s\in\mathbb{Z}} i^s (1 - e^{-\frac{1}{2}\pi i s \Delta})^k (1 + e^{-\frac{1}{2}\pi i s \Delta})^{d-k} \hat{\tilde{\Phi}}(s)$$

where

$$\begin{aligned}
\hat{\tilde{\Phi}}(s) &= \frac{1}{4} \int_{-2}^{2} \tilde{\Phi}(t) e^{-\frac{1}{2}\pi i s t} \,\mathrm{d}t \\
&= \frac{1}{4} \int_{-1}^{1} \Phi(t) (e^{-\frac{1}{2}\pi i s t} + (-1)^s e^{\frac{1}{2}\pi i s t}) \,\mathrm{d}t \\
&= \begin{cases} \frac{1}{2} \int_{-1}^{1} \Phi(t) \cos(\frac{1}{2}\pi s t) \,\mathrm{d}t & \textit{if } s \textit{ is even} \\ -\frac{i}{2} \int_{-1}^{1} \Phi(t) \sin(\frac{1}{2}\pi s t) \,\mathrm{d}t & \textit{if } s \textit{ is odd} \end{cases}
\end{aligned}$$

denote the Fourier coefficients of $\tilde{\Phi}$ on $[-2, 2]$. (Here i is the imaginary unit here, not an index).

Recovering the values of Φ given the eigenvalues $\mu_0, \ldots, \mu_d$. Conversely, given eigenvalues $\mu_0, \ldots, \mu_d$ corresponding to each monomial degree, we can recover the entries of the matrix K.

Theorem F.6. *For any $x, y \in \square^d$ with Hamming distance r,*

$$K(x, y) = \Phi\left(\left(\frac{d}{2} - r\right)\Delta\right) = \sum_{k=0}^{d} C_k^{d-r,r} \mu_k,$$

where $C_k^{d-r,r} = \sum_{j=0} (-1)^{k+j} \binom{d-r}{j} \binom{r}{k-j}$ as in Eq. (7).

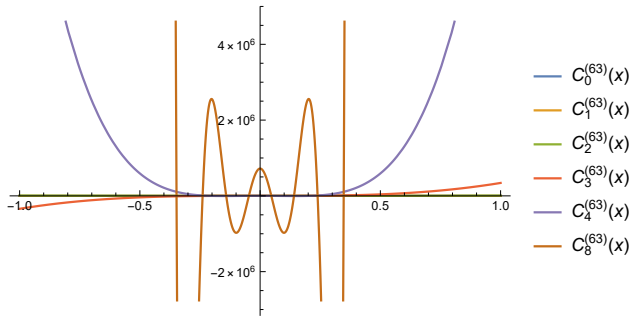


Figure 10: Examples of Gegenbauer Polynomials for $d = 128$ (or $\alpha = 63$).

Proof. Recall that for any $S \subseteq [d]$, $\chi_S(x) = x^S$ is the Fourier basis corresponding to S (see Eq. (3)). Then by converting from the Fourier basis to the regular basis, we get

$$\Phi\left(\left(\frac{d}{2} - r\right)\Delta\right) = K(x, y) \quad \text{for any } x, y \in \square^d \text{ with Hamming distance } r$$

$$= \sum_{k=0}^{d} \mu_k \sum_{|S|=k} \chi_S(x)\chi_S(y).$$

If x and y differ on a set $T \subseteq [d]$, then we can simplify the inner sum

$$\Phi\left(\left(\frac{d}{2} - r\right)\Delta\right) = \sum_{k=0}^{d} \mu_k \sum_{|S|=k} (-1)^{|S \cap T|} = \sum_{k=0}^{d} \mu_k C_k^{d-r,r}.$$

□

Remark F.7. If we let $\mathcal{T}$ be the operator that sends $\mu_\bullet \mapsto \mu_{\bullet+1}$, then we have the following operator expression

$$\Phi\left(\left(\frac{d}{2} - r\right)\Delta\right) = [(1 + \mathcal{T})^{d-r}(1 - \mathcal{T})^r \mu]_0$$

Remark F.8. The above shows that the matrix $C = \{C_k^{d-r,r}\}_{k,r=0}^{d}$ satisfies

$$C^2 = 2^d I.$$

F.2 Sphere

Now let's consider the case when $\mathcal{X} = \sqrt{d}\mathcal{S}^{d-1}$ is the radius-$\sqrt{d}$ sphere in $\mathbb{R}^d$ equipped with the uniform measure. Again, because $x \in \mathcal{X}$ all have the same norm, we will consider Φ as a univariate function with $K(x, y) = \Phi(\langle x, y\rangle / \|x\|\|y\|) = \Phi(\langle x, y\rangle / d)$.

As is long known (Schoenberg, 1942; Gneiting, 2013; Xu and Cheney, 1992; Smola et al., 2001), K is diagonalized by spherical harmonics. We review these results briefly below, as we will build on them to deduce spectral information of K on isotropic Gaussian distributions.

Review: spherical harmonics and Gegenbauer polynomials. *Spherical harmonics* are L^2 functions on $\mathcal{S}^{d-1}$ that are eigenfunctions of the Laplace-Beltrami operator $\Delta_{\mathcal{S}^{d-1}}$ of $\mathcal{S}^{d-1}$. They can be described as the restriction of certain homogeneous polynomials in $\mathbb{R}^d$ to $\mathcal{S}^{d-1}$. Denote by $\mathcal{H}^{d-1,(l)}$ the space of spherical harmonics of degree l on sphere S^{d-1}. Then we have the orthogonal decomposition $L^2(\mathcal{S}^{d-1}) \cong \bigoplus_{l=0}^{\infty} \mathcal{H}^{d-1,(l)}$. It is a standard fact that $\dim \mathcal{H}^{d-1,(l)} = \binom{d-1+l}{d-1} - \binom{d-3+l}{d-1}$.

There is a special class of spherical harmonics called *zonal harmonics* that can be represented as $x \mapsto p(\langle x, y\rangle)$ for specific polynomials $p : \mathbb{R} \to \mathbb{R}$, and that possess a special *reproducing property* which we will describe shortly. Intuitively, the value of any zonal harmonics only depends on the "height" of x along some fixed axis y, so a typical zonal harmonics looks like Fig. 11. The polynomials p must be one of the Gegenbauer polynomials. *Gegenbauer polynomials* $\{C_l^{(\alpha)}(t)\}_{l=0}^{\infty}$ are orthogonal

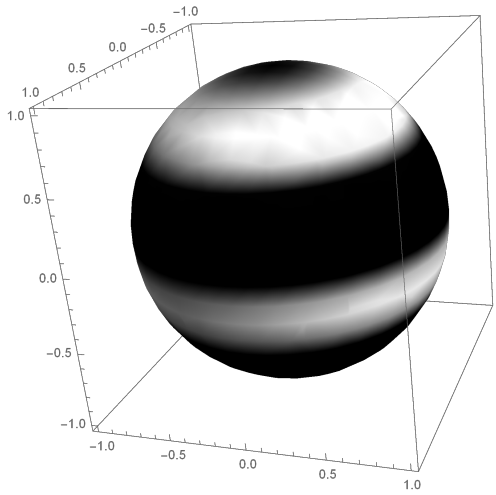


Figure 11: Visualization of a zonal harmonic, which depends only on the "height" of the input along a fixed axis. Color indicates function value.

polynomials with respect to the measure $(1-t^2)^{\alpha-\frac{1}{2}}$ on $[-1,1]$ (see Fig. 10 for examples), and here we adopt the convention that

$$\int_{-1}^{1} C_n^{(\alpha)}(t)C_l^{(\alpha)}(t)(1-t^2)^{\alpha-\frac{1}{2}}\ \mathrm{d}t = \frac{\pi 2^{1-2\alpha}\Gamma(n+2\alpha)}{n!(n+\alpha)[\Gamma(\alpha)]^2}\mathbb{I}(n=l).$$

Then for each (oriented) axis $y \in \mathcal{S}^{d-1}$ and degree l, there is a unique zonal harmonic

$$Z_y^{d-1,(l)} \in \mathcal{H}^{d-1,(l)}, \quad Z_y^{d-1,(l)}(x) \stackrel{\text{def}}{=} c_{d,l}^{-1} C_l^{(\frac{d-2}{2})}(\langle x, y\rangle)$$

for any $x, y \in \mathcal{S}^{d-1}$, where $c_{d,l} = \frac{d-2}{d+2l-2}$. Very importantly, they satisfy the following

Fact F.9 (Reproducing property (Suetin))**.** *For any $f \in \mathcal{H}^{d-1,(m)}$,*

$$\mathop{\mathbb{E}}_{z\sim\mathcal{S}^{d-1}} Z_y^{d-1,(l)}(z)f(z) = f(y)\mathbb{I}(l=m)$$

$$\mathop{\mathbb{E}}_{z\sim\mathcal{S}^{d-1}} Z_y^{d-1,(l)}(z)Z_x^{d-1,(m)}(z) = Z_y^{d-1,(l)}(x)\mathbb{I}(l=m) = c_{d,l}^{-1}C_l^{(\frac{d-2}{2})}(\langle x,y\rangle)\mathbb{I}(l=m)$$

Theorem F.10 (Schoenberg)**.** *Suppose $\Phi : [-1,1] \to \mathbb{R}$ is in $L^2((1-t^2)^{\frac{d-1}{2}-1})$, so that it has the Gegenbauer expansion*

$$\Phi(t) \stackrel{a.e.}{=} \sum_{l=0}^{\infty} a_l c_{d,l}^{-1} C_l^{(\frac{d-2}{2})}(t).$$

Then K has eigenspaces $\mathcal{H}_{\sqrt{d}}^{d-1,(l)} \stackrel{\text{def}}{=} \{f(x/\sqrt{d}) : f \in \mathcal{H}^{d-1,(l)}\}$ with corresponding eigenvalues a_l. Since $\bigoplus_{l=0}^{\infty}\mathcal{H}_{\sqrt{d}}^{d-1,(l)}$ is an orthogonal decomposition of $L^2(\sqrt{d}\mathcal{S}^{d-1})$, this describes all eigenfunctions of K considered as an operator on $L^2(\sqrt{d}\mathcal{S}^{d-1})$.

For completeness, we include the proof of this theorem in Appendix G.

F.3 Isotropic Gaussian

Now let's consider $\mathcal{X} = \mathbb{R}^d$ equipped with standard isotropic Gaussian $\mathcal{N}(0, I)$, so that K behaves like

$$Kf(x) = \mathop{\mathbb{E}}_{y\sim\mathcal{N}(0,I)} K(x,y)f(y) = \mathop{\mathbb{E}}_{y\sim\mathcal{N}(0,I)} \Phi\left(\frac{\langle x,y\rangle}{\|x\|\|y\|}, \frac{\|x\|^2}{d}, \frac{\|y\|^2}{d}\right) f(y)$$

for any $f \in L^2(\mathcal{N}(0,I))$. In contrast to the previous two sections, K will essentially depend on the effect of the norms $\|x\|$ and $\|y\|$ on Φ.

Note that an isotropic Gaussian vector $z \sim \mathcal{N}(0, I)$ can be sampled by independently sampling its direction v uniformly from the sphere $\mathcal{S}^{d-1}$ and sampling its magnitude r from a chi distribution χ_d with d degrees of freedom. Proceeding along this line of logic yields the following spectral theorem:

Theorem F.11. *A function $K : (\mathbb{R}^d)^2 \to \mathbb{R}$ of the form $K(x, y) = \Phi(\langle x, y\rangle/\|x\|\|y\|, \|x\|/d, \|y\|/d)$ forms a positive semidefinite Hilbert-Schmidt operator on $L^2(\mathcal{N}(0, I))$ iff Φ can be decomposed as*

$$\Phi(t, q, q') = \sum_{l=0}^{\infty} A_l(q, q') c_{d,l}^{-1} C_l^{(\frac{d-2}{2})}(t) \tag{12}$$

where $C_l^{(\frac{d-2}{2})}(t)$ are Gegenbauer polynomials as in Appendix F.2 and A_l are positive semidefinite Hilbert-Schmidt kernels on $L^2(\frac{1}{d}\chi_d^2)$, the L^2 space over the probability measure of a χ_d^2-variable divided by d. In addition, K is positive definite iff all A_l are.

For certain simple F, we can obtain $\{A_l\}_{l\geq 0}$ explicitly. For example, suppose K is *degree-s positive-homogeneous*, in the sense that, for $a, b > 0$,

$$K(ax, by) = (ab)^s K(x, y).$$

This happens when K is the CK or NTK of an MLP with degree-s positive-homogeneous. Then it's easy to see that $\Phi(t, q, q') = (qq')^s \bar{\Phi}(t)$ for some $\bar{\Phi} : [-1, 1] \to \mathbb{R}$, and

$$\Phi(t, q, q') = \sum_{l=0}^{\infty} (qq')^s a_l c_{d,l}^{-1} C_l^{(\frac{d-2}{2})}(t)$$

where $\{a_l\}_l$ are the Gegenbauer coefficients of $\bar{\Phi}$,

$$\bar{\Phi}(t) = \sum_{l=0}^{\infty} a_l c_{d,l}^{-1} C_l^{(\frac{d-2}{2})}(t).$$

We can then conclude with the following theorem.

Theorem F.12. *Suppose $K : (\mathbb{R}^d)^2 \to \mathbb{R}$ is a kernel given by*

$$K(x, y) = R(\|x\|/d) R(\|x\|/d) \bar{\Phi}(\langle x, y\rangle/\|x\|\|y\|)$$

for some functions $R : [0, \infty) \to \mathbb{R}$, $\bar{\Phi} : [-1, 1] \to \mathbb{R}$. Let

$$\bar{\Phi}(t) = \sum_{l=0}^{\infty} a_l c_{d,l}^{-1} C_l^{(\frac{d-2}{2})}(t)$$

be the Gegenbauer expansion of $\bar{\Phi}$. Also define

$$\lambda \stackrel{\text{def}}{=} \sqrt{\mathop{\mathbb{E}}_{q\sim\frac{1}{d}\chi_d^2} R(q)^2}.$$

Then over the standard Gaussian in $\mathbb{R}^d$, K has the following eigendecomposition

- *For each $l \geq 0$,*

  $$\lambda^{-1} R \otimes \mathcal{H}^{d-1,(l)} = \{\lambda^{-1} R(\|x\|^2/d) f(x/\|x\|) : f \in \mathcal{H}^{d-1,(l)}\}$$

  is an eigenspace with eigenvalue $\lambda^2 a_l$.

- *For any $S \in L^2(\frac{1}{d}\chi_d^2)$ that is orthogonal to R, i.e.*

  $$\mathop{\mathbb{E}}_{q\sim\frac{1}{d}\chi_d^2} S(q) R(q) = 0,$$

  the function

  $$S(\|x\|^2/d) f(x/\|x\|)$$

  for any $f \in L^2(\mathcal{S}^{d-1})$ is in the null space of K.

Proof. The A_l in Eq. (12) for K are all equal to

$$A_l(q, q') = \lambda^2 \frac{R(q)}{\lambda} \frac{R(q')}{\lambda}.$$

This is a rank 1 kernel (on $L^2(\frac{1}{d}\chi_d^2)$), with eigenfunction R/λ and eigenvalue λ^2. The rest then follows straightforwardly from Thm F.10. □

A common example where Thm F.12 applies is when K is the CK of an MLP with relu, or more generally degree-s positive homogeneous activation functions, so that the R in Thm F.12 is a polynomial.

In general, we cannot expect K can be exactly diagonalized in a natural basis, as $\{A_l\}_{l\geq 0}$ cannot even be simultaneously diagonalizable. We can, however, investigate the "variance due to each degree of spherical harmonics" by computing

$$a_l \stackrel{\text{def}}{=} \mathop{\mathbb{E}}_{q,q' \sim d^{-1}\chi_d^2} A_l(q, q')$$

which is the coefficient of Gegenbauer polynomials in

$$\hat{\Phi}_d(t) \stackrel{\text{def}}{=} \mathop{\mathbb{E}}_{q,q' \sim d^{-1}\chi_d^2} \Phi(t, q, q') = \sum_{l=0}^{\infty} a_l c_{d,l}^{-1} C_l^{(\frac{d-2}{2})}(t). \tag{13}$$

F.4 In High Dimension, Boolean Cube ≈ Sphere ≈ Standard Gaussian

For the same neural kernel K with $K(x, y) = \Phi(\langle x, y\rangle/\|x\|\|y\|, \|x\|^2/d, \|y\|^2/d)$, let $\hat{\Phi}_d$ be as defined in Eq. (13), and let $\Phi_{\mathcal{S}} \stackrel{\text{def}}{=} \Phi(t, 1, 1)$. Thus $\Phi_{\mathcal{S}}$ is the univariate Φ that we studied in Appendix F.2 in the context of the sphere.

As $d \to \infty$, $\frac{1}{d}\chi_d^2$ converges almost surely to 1, and each A_l in Eq. (12) converges weakly to the simple operator that multiplies the input function by $A_l(1, 1)$. We verify in Fig. 12 that, for different erf kernels, $\hat{\Phi}_d$ approximates $\Phi_{\mathcal{S}}$ when d is large, which would then imply that their spectra become identical for large d. Note that this is tautologically true for relu kernels with no bias $\sigma_b = 0$ because they are positive-homogeneous.

Next, we compute the eigenvalues of K on the boolean cube and the sphere, as well as the eigenvalues of the kernel $\hat{K}(x, y) \stackrel{\text{def}}{=} \hat{\Phi}_d(\langle x, y\rangle/\|x\|\|y\|)$ on the sphere, which "summarizes" the eigenvalues of K on the standard Gaussian distribution. In Fig. 13, we compare them for different dimensions d up to degree 5 (there, "Gaussian" means $\hat{K}$ on the sphere). We see that by dimension 128, all eigenvalues shown are very close to each other for each data distribution.

Finally, we note that a precise, mathematical statement can be made about the how the kernel behaves the same over the boolean cube and over the sphere, by leveraging the *invariance principle* (O'Donnell, 2014). However, here we are content with numerical evidence and leave this for future work.

G Omitted Proofs

Theorem 3.1. *On the d-dimensional boolean cube $\square^d$, for every $S \subseteq [d]$, χ_S is an eigenfunction of K with eigenvalue*

$$\mu_{|S|} \stackrel{\text{def}}{=} \mathop{\mathbb{E}}_{x \in \square^d} x^S K(x, \mathbb{1}) = \mathop{\mathbb{E}}_{x \in \square^d} x^S \Phi\left(\sum_i x_i/d\right), \tag{4}$$

where $\mathbb{1} = (1, \ldots, 1) \in \square^d$. This definition of $\mu_{|S|}$ does not depend on the choice S, only on the cardinality of S. These are all of the eigenfunctions of K by dimensionality considerations.[7]

[7] Readers familiar with boolean Fourier analysis may be reminded of the *noise operator* T_ρ, $\rho \leq 1$ (O'Donnell, 2014, Defn 2.46). In the language of this work, T_ρ is a neural kernel with eigenvalues $\mu_k = \rho^k$.

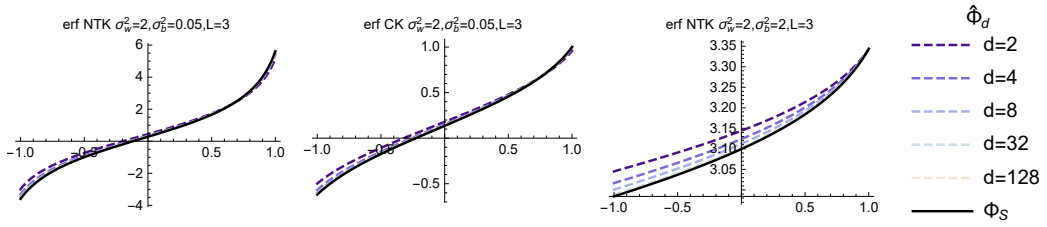


Figure 12: **In high dimension d, $\hat{\Phi}_d$ as defined in Eq. (13) approximates Φ very well.** We show for 3 different erf kernels that the function Φ defining K as an integral operator on the sphere is well-approximated by $\hat{\Phi}_d$ when d is moderately large ($d \geq 32$ seems to work well). This suggests that the eigenvalues of K as an integral operator on the standard Gaussian should approximate those of K as an integral operator on the sphere.

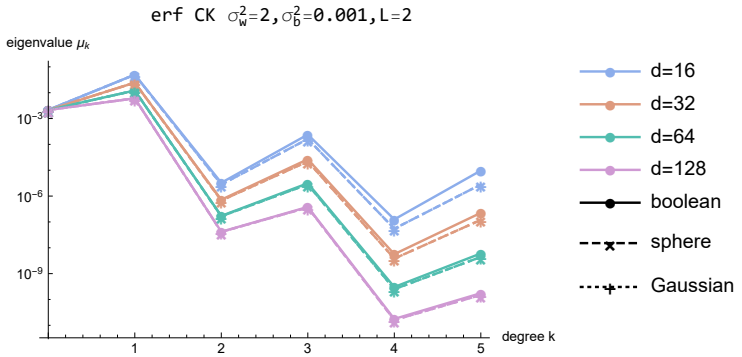


Figure 13: **In high dimension d, the eigenvalues are very close for the kernel over the boolean cube, the sphere, and standard Gaussian.** We plot the eigenvalues μ_k of the erf CK, with $\sigma_w^2 = 2, \sigma_b^2 = 0.001$, depth 2, over the boolean cube, the sphere, as well as kernel on the sphere induced by $\hat{\Phi}_d$ (Eq. (13)). We do so for each degree $k \leq 5$ and for dimensions $d = 16, 32, 64, 128$. We see that by dimension $d = 128$, the eigenvalues shown are already very close to each other.

Proof. We directly verify $K\chi_S = \mu_{|S|}\chi_S$. Notice first that

$$K(x,y) = \Phi(\langle x, y\rangle) = \Phi(\langle x \odot y, \mathbb{1}\rangle) = K(x \odot y, \mathbb{1})$$

where $\odot$ is Hadamard product. We then calculate

$$\begin{aligned} K\chi_S(y) &= \mathop{\mathbb{E}}_{x} K(y,x)x^S \\ &= \mathop{\mathbb{E}}_{x} K(\mathbb{1}, x \odot y)(x \odot y)^S y^S. \end{aligned}$$

Here we are using the fact that x and y are boolean to get $x^S = (x \odot y)^S y^S$. Changing variable $z \stackrel{\text{def}}{=} x \odot y$, we get

$$K\chi_S(y) = y^S \mathop{\mathbb{E}}_{z} K(\mathbb{1}, z) z^S = \mu_{|S|}\chi_S(y)$$

as desired. Finally, note that $\mu_{|S|}$ is invariant under permutation of $[d]$, so indeed it depends only on the size of S. □

Theorem F.10 (Schoenberg)**.** *Suppose $\Phi : [-1,1] \to \mathbb{R}$ is in $L^2((1-t^2)^{\frac{d-1}{2}-1})$, so that it has the Gegenbauer expansion*

$$\Phi(t) \stackrel{a.e.}{=} \sum_{l=0}^{\infty} a_l c_{d,l}^{-1} C_l^{(\frac{d-2}{2})}(t).$$

Then K has eigenspaces $\mathcal{H}^{d-1,(l)}_{\sqrt{d}} \stackrel{\text{def}}{=} \{f(x/\sqrt{d}) : f \in \mathcal{H}^{d-1,(l)}\}$ with corresponding eigenvalues a_l. Since $\bigoplus_{l=0}^{\infty} \mathcal{H}^{d-1,(l)}_{\sqrt{d}}$ is an orthogonal decomposition of $L^2(\sqrt{d}\mathcal{S}^{d-1})$, this describes all eigenfunctions of K considered as an operator on $L^2(\sqrt{d}\mathcal{S}^{d-1})$.

Proof. Let $f \in \mathcal{H}^{d-1,(n)}$. Then for any $x \in \sqrt{d}\mathcal{S}^{d-1}$,

$$\begin{aligned} \mathop{\mathbb{E}}_{y \in \sqrt{d}\mathcal{S}^{d-1}} K(x,y) f(y/\sqrt{d}) &= \mathop{\mathbb{E}}_{y \in \sqrt{d}\mathcal{S}^{d-1}} \Phi(\langle x, y\rangle/d) f(y/\sqrt{d}) \\ &= \mathop{\mathbb{E}}_{\tilde{y} \in \mathcal{S}^{d-1}} \Phi(\langle x/\sqrt{d}, \tilde{y}\rangle) f(\tilde{y}) \\ &= \mathop{\mathbb{E}}_{\tilde{y} \in \mathcal{S}^{d-1}} \left(\sum_{l=0}^{\infty} a_l \frac{1}{c_{d,l}} C_l^{(\frac{d-2}{2})}(\langle x/\sqrt{d}, \tilde{y}\rangle) \right) f(\tilde{y}) \\ &= \mathop{\mathbb{E}}_{\tilde{y} \in \mathcal{S}^{d-1}} \left(\sum_{l=0}^{\infty} a_l Z^{d-1,(l)}_{x/\sqrt{d}}(\tilde{y}) \right) f(\tilde{y}) \\ &= a_n \mathop{\mathbb{E}}_{\tilde{y} \in \mathcal{S}^{d-1}} Z^{d-1,(n)}_{x/\sqrt{d}}(\tilde{y}) f(\tilde{y}) \qquad \text{by orthogonality} \\ &= a_n f(x/\sqrt{d}) \end{aligned}$$

by reproducing property. □

Lemma 3.2. *With μ_k as in Thm 3.1,*

$$\mu_k = 2^{-d}(I - \mathcal{T}_\Delta)^k (I + \mathcal{T}_\Delta)^{d-k} \Phi(1) \tag{5}$$

$$= 2^{-d} \sum_{r=0}^{d} C_r^{d-k,k} \Phi\left(\left(\frac{d}{2} - r \right) \Delta \right) \tag{6}$$

where

$$C_r^{d-k,k} \stackrel{\text{def}}{=} \sum_{j=0} (-1)^{r+j} \binom{d-k}{j} \binom{k}{r-j}. \tag{7}$$

Proof. Because $\sum_i x_i/d$ only takes on values $\{-\frac{d}{2}\Delta, (-\frac{d}{2}+1)\Delta, \ldots, (\frac{d}{2}-1)\Delta, \frac{d}{2}\Delta\}$, where $\Delta = \frac{2}{d}$, we can collect like terms in Eq. (4) and obtain

$$\mu_k = 2^{-d} \sum_{r=0}^{d} \sum_{x \text{ has } r \text{ '}-1\text{'s}} \left(\prod_{i=1}^{k} x_i \right) \Phi\left(\left(\frac{d}{2} - r \right) \Delta \right)$$

which can easily be shown to be equal to

$$\mu_k = 2^{-d} \sum_{r=0}^{d} C_r^{d-k,k} \Phi\left(\left(\frac{d}{2} - r \right) \Delta \right),$$

proving Eq. (6) in the claim. Finally, observe that $C_r^{d-k,k}$ is also the coefficient of x^r in the polynomial $(1-x)^k(1+x)^{d-k}$. Some operator arithmetic then yields Eq. (5). □

Theorem F.11. *A function $K : (\mathbb{R}^d)^2 \to \mathbb{R}$ of the form $K(x,y) = \Phi(\langle x,y\rangle/\|x\|\|y\|, \|x\|/d, \|y\|/d)$ forms a positive semidefinite Hilbert-Schmidt operator on $L^2(\mathcal{N}(0,I))$ iff Φ can be decomposed as*

$$\Phi(t,q,q') = \sum_{l=0}^{\infty} A_l(q,q') c_{d,l}^{-1} C_l^{(\frac{d-2}{2})}(t) \tag{12}$$

where $C_l^{(\frac{d-2}{2})}(t)$ are Gegenbauer polynomials as in Appendix F.2 and A_l are positive semidefinite Hilbert-Schmidt kernels on $L^2(\frac{1}{d}\chi_d^2)$, the L^2 space over the probability measure of a χ_d^2-variable divided by d. In addition, K is positive definite iff all A_l are.

Proof. Note that an isotropic Gaussian vector $z \sim \mathcal{N}(0,I)$ can be sampled by independently sampling its direction v uniformly from the sphere $\mathcal{S}^{d-1}$ and sampling its magnitude r from a chi distribution χ_d with d degrees of freedom. In the following, we adopt the following notations

$$x, y \in \mathbb{R}^d,\ q = \|x\|^2/d,\ q' = \|y\|^2/d,\ v = x/\|x\|,\ v' = y/\|y\|$$

Then, by the reasoning above,

$$Kf(x/\sqrt{d}) = \mathop{\mathbb{E}}_{\substack{v' \sim \mathcal{S}^{d-1} \\ q' \sim d^{-1}\chi_d^2}} \Phi\left(\langle v, v'\rangle, q, q'\right) f(\sqrt{q'}v).$$

Here $f \in L^2(\mathcal{N}(0, I/d))$ (so that $f(\cdot/\sqrt{d}) \in L^2(\mathcal{N}(0,I))$) and $d^{-1}\chi_d^2$ is the distribution of a χ_d^2 random variable divided by d. Because of this decomposition of $\mathcal{N}(0,I)$ into a product distribution of direction and magnitude, its space of L^2 functions naturally decomposes into a tensor product of corresponding L^2 spaces

$$L^2(\mathcal{N}(0,I)) = L^2(\mathcal{S}^{d-1}) \times L^2(d^{-1}\chi_d^2).$$

where every $f \in L^2(\mathcal{N}(0,I))$ can be written as a sum

$$f(x) = \sum_{l=0}^{\infty} R_l(q) P_l(v)$$

where $q = \|x\|^2/d, v = x/\|x\|$ as above, $P_l \in \mathcal{H}^{d-1,(l)}$ is a spherical harmonic of degree l, and $R_l \in L^2(d^{-1}\chi_d^2)$ (i.e. $R_l(\cdot/d) \in L^2(\chi_d^2)$).

Likewise, assuming K is Hilbert-Schmidt, then its Hilbert-Schmidt norm (measured against the uniform distribution over the sphere) is bounded:

$$\begin{aligned} \|K\|_{HS}^2 &= \mathop{\mathbb{E}}_{\substack{v,v' \sim \mathcal{S}^{d-1} \\ q,q' \sim d^{-1}\chi_d^2}} \Phi(\langle v, v'\rangle, q, q')^2 \\ &= \mathop{\mathbb{E}}_{q,q' \sim d^{-1}\chi_d^2} \int_{-1}^{1} \Phi(t,q,q')^2 (1-t^2)^{\frac{d-1}{2}-1} \,\mathrm{d}t. \end{aligned}$$

Thus Φ resides in the tensor product space

$$\Phi \in L^2((1-t^2)^{\frac{d-1}{2}-1}) \otimes L^2(d^{-1}\chi_d^2) \otimes L^2(d^{-1}\chi_d^2)$$

and therefore can be expanded as

$$\Phi(t, q, q') = \sum_{l=0}^{\infty} A_l(q, q') c_{d,l}^{-1} C_l^{(\frac{d-2}{2})}(t)$$

where $C_l^{(\frac{d-2}{2})}(t)$ are Gegenbauer polynomials as in Appendix F.2 and $A_l \in L^2(d^{-1}\chi_d^2)^{\otimes 2}$. By Lem G.1 below, K being Hilbert-Schmidt implies that each A_l is, as well. Conversely, if each A_l is Hilbert-Schmidt, then K is obviously a Hilbert-Schmidt kernel.

□

Lemma G.1. *For each l, A_l is a positive semidefinite Hilbert-Schmidt kernel. It is positive definite if K is.*

Proof. With $q = \|x\|^2/d, v = x/\|x\|$ as above, let $f(x/\sqrt{d}) = f(\sqrt{q}v) = R_m(q)P_m(v)$ for some degree m nonzero spherical harmonics $P_m \in \mathcal{H}^{d-1,(m)}$ and some scalar function $R_m \in L^2(d^{-1}\chi_d^2)$ that is not a.e. zero. We have

$$\begin{aligned}
K\phi(x/\sqrt{d}) &= \mathop{\mathbb{E}}_{\substack{v' \sim \mathcal{S}^{d-1} \\ q' \sim d^{-1}\chi_d^2}} \Phi(\langle v, v' \rangle, q, q') f(\sqrt{q'}v) \\
&= \mathop{\mathbb{E}}_{\substack{v' \sim \mathcal{S}^{d-1} \\ q' \sim d^{-1}\chi_d^2}} R_m(q')P_m(v') \sum_{l=0}^{\infty} A_l(q, q') c_{d,l}^{-1} C_l^{(\frac{d-2}{2})}(\langle v, v' \rangle) \\
&= \mathop{\mathbb{E}}_{\substack{v' \sim \mathcal{S}^{d-1} \\ q' \sim d^{-1}\chi_d^2}} R_m(q')P_m(v') \sum_{l=0}^{\infty} A_l(q, q') Z_v^{d-1,(l)}(v') \\
&= P_m(v) \mathop{\mathbb{E}}_{q' \sim d^{-1}\chi_d^2} A_m(q, q') R_m(q').
\end{aligned}$$

Therefore,

$$\mathop{\mathbb{E}}_{x \sim \mathcal{N}(0,I)} f(x) K f(x/\sqrt{d}) = \left(\mathop{\mathbb{E}}_{v \sim \mathcal{S}^{d-1}} P_m(v)^2 \right) \left(\mathop{\mathbb{E}}_{q, q' \sim d^{-1}\chi_d^2} R_m(q) A_m(q, q') R_m(q') \right)$$

is nonnegative, and is positive if K is positive definite, by the assumption above that f is not a.e. zero. Since $\mathbb{E}\, P_m(v)^2 > 0$ as $P_m \neq 0$, we must have

$$\mathop{\mathbb{E}}_{q, q' \sim d^{-1}\chi_d^2} R_m(q) A_m(q, q') R_m(q') \geq 0 \text{ (or } > 0 \text{ if } K \text{ is positive definite).}$$

This argument holds for any $R_m \in L^2(d^{-1}\chi_d^2)$ that is not a.e. zero, so this implies that A_l is positive semidefinite, and positive definite if K is so. □

Weak Spectral Simplicity Bias

Theorem 4.1 (Weak Spectral Simplicity Bias)**.** *Let K be the CK or NTK of an MLP on a boolean cube $\square^d$. Then the eigenvalues $\mu_k, k = 0, \ldots, d$, satisfy*

$$\mu_0 \geq \mu_2 \geq \cdots \geq \mu_{2k} \geq \cdots, \quad \mu_1 \geq \mu_3 \geq \cdots \geq \mu_{2k+1} \geq \cdots. \tag{9}$$

Proof. Again, there is a function Φ such that $K(x, y) = \Phi(\langle x, y \rangle / \|x\|\|y\|)$. By Thm G.3, Φ has a Taylor expansion with only nonnegative coefficients.

$$\Phi(c) = a_0 + a_1 c + a_2 c^2 + \cdots.$$

By Lem G.2, Eq. (9) is true for polynomials $\Phi(c) = c^r$,

$$\mu_0(c^r) \geq \mu_2(c^r) \geq \cdots, \quad \mu_1(c^r) \geq \mu_3(c^r) \geq \cdots. \tag{14}$$

Then since each $\mu_k = \mu_k(\Phi)$ is a linear function of Φ, $\mu_k(\Phi) = \sum_{r=0}^{\infty} a_r \mu_k(c^r)$ also follows the same ordering. □

Lemma G.2. *Let $\mathcal{T}_\Delta$ be the shift operator with step Δ that sends a function $\Phi(\cdot)$ to $\Phi(\cdot - \Delta)$. Let $\Phi(c) = c^t$ for some t. Let μ_k^d be the eigenvalue of $K(x, y) = \Phi(\langle x, y\rangle / \|x\|\|y\|)$ on the boolean cube $\square^d$. Then for any $0 \le k \le d$,*

$$\mu_k^d = 0 \qquad \text{if } t + k \text{ is odd} \tag{15}$$

$$1 \ge \mu_{k-2}^d \ge \mu_k^d \ge 0 \qquad \text{if } t + k \text{ is even.} \tag{16}$$

We furthermore have the identity

$$\mu_{k-2}^d - \mu_k^d = \left(\frac{d-2}{d}\right)^t \mu_{k-2}^{d-2}$$

for any $2 \le k \le d$.

Proof. As in Eq. (5),

$$2^{-d}(I - \mathcal{T}_\Delta)^k (I + \mathcal{T}_\Delta)^{d-k}\Phi(1) = 2^{-d}\sum_{r=0}^{d} C_r^{d-k,k}\Phi\left(\left(\frac{d}{2} - r\right)\Delta\right)$$

where $C_r^{d-k,k} \stackrel{\text{def}}{=} \sum_{j=0}(-1)^{r+j}\binom{d-k}{j}\binom{k}{r-j}$. It's easy to observe that

$$C_r^{d-k,k} = C_{d-r}^{d-k,k} \quad \text{if } k \text{ is even}$$
$$C_r^{d-k,k} = -C_{d-r}^{d-k,k} \quad \text{if } k \text{ is odd.}$$

In the first case, when $\Phi(c) = c^t$ with t odd, then by symmetry $\mu_k = 0$. Similarly for the second case when t is even. This finishes the proof for Eq. (15).

Note that it's clear from the form of Eq. (5) that $\mu_k^d \le \Phi(1) = 1$ always. So we show via induction on d that the rest of Eq. (16) is true for any $t \in \mathbb{N}$. The induction will reduce the case of d to the case of $d - 2$. So for the base case, we need both $d = 0$ and $d = 1$. Both can be shown by some simple calculations.

Now for the inductive step, assume Eq. (16) for $d - 2$, and we observe that

$$\begin{aligned}
\mu_{k-2}^d - \mu_k^d &= 2^{-d}(I - \mathcal{T}_{2/d})^{k-2}(I + \mathcal{T}_{2/d})^{d-k}((I + \mathcal{T}_{2/d})^2 - (I - \mathcal{T}_{2/d})^2)\Phi(1) \\
&= 2^{-d}(I - \mathcal{T}_{2/d})^{k-2}(I + \mathcal{T}_{2/d})^{d-k}(4\mathcal{T}_{2/d})\Phi(1) \\
&= 2^{-d+2}(I - \mathcal{T}_{2/d})^{k-2}(I + \mathcal{T}_{2/d})^{d-k}\Phi(1 - 2/d).
\end{aligned}$$

By Eq. (5), we can expand this into

$$\begin{aligned}
&2^{-d+2}\sum_{r=0}^{d-2} C_r^{d-k,k-2}\Phi\left(1 - \frac{2}{d} - r\frac{2}{d}\right) \\
&= 2^{-d+2}\sum_{r=0}^{d-2} C_r^{d-k,k-2}\left(1 - \frac{2}{d} - r\frac{2}{d}\right)^t \\
&= \left(\frac{d-2}{d}\right)^t 2^{-d+2}\sum_{r=0}^{d-2} C_r^{d-k,k-2}\left(1 - r\frac{2}{d-2}\right)^t \\
&= \left(\frac{d-2}{d}\right)^t \mu_{k-2}^{d-2}.
\end{aligned}$$

By induction, $\mu_{k-2}^{d-2} \ge 0$, so we have

$$\mu_{k-2}^d - \mu_k^d = \left(\frac{d-2}{d}\right)^t \mu_{k-2}^{d-2} \ge 0$$

as desired.

□

Theorem G.3. *Let K be the CK or NTK of an MLP with domain $\sqrt{d}\mathcal{S}^{d-1} \subseteq \mathbb{R}^d$. Then $K(x, y) = \Phi\left(\frac{\langle x,y\rangle}{\|x\|\|y\|}\right)$ where $\Phi : [-1, 1] \to \mathbb{R}$ has a Taylor series expansion around 0*

$$\Phi(c) = a_0 + a_1 c + a_2 c^2 + \cdots \tag{17}$$

with the properties that $a_i \geq 0$ for all i and $\sum_i a_i = \Phi(1) \leq \infty$, so that Eq. (17) is absolutely convergent on $c \in [-1, 1]$.

Proof. We first prove this statement for the CK of an L layer MLP with nonlinearity ϕ. Let Φ^l be the corresponding Φ-function for the CK Σ^l. It suffices to prove this by induction on depth L. For $L = 1$, $\Sigma^1(x, x') = \sigma_w^2(n^0)^{-1}\langle x, x'\rangle + \sigma_b^2$ by Eq. (CK), so clearly $\Phi^1(c) = a_0 + a_1 c$ where a_0 and a_1 are nonnegative.

Now for the inductive step, suppose Σ^{l-1} satisfies the property that Φ^{l-1} has Taylor expansion of the desired form. We seek to show the same for Σ^l and Φ^l. First notice that it suffices to show this for $\mathrm{V}_\phi(\Sigma^{l-1})$, since multiplication by σ_w^2 and addition of σ_b^2 preserves the property. But

$$\mathrm{V}_\phi(\Sigma^{l-1})(x, x') = \mathbb{E}\,\phi(z)\phi(z')$$

where

$$(z, z') \sim \mathcal{N}\left(0, \begin{pmatrix} \Sigma^{l-1}(x,x) & \Sigma^{l-1}(x,x') \\ \Sigma^{l-1}(x,x') & \Sigma^{l-1}(x',x') \end{pmatrix}\right) = \mathcal{N}\left(0, \begin{pmatrix} \Phi^{l-1}(1) & \Phi^{l-1}(c) \\ \Phi^{l-1}(c) & \Phi^{l-1}(1) \end{pmatrix}\right).$$

Using the notation of Lem G.4, we can express this as

$$\mathrm{V}_\phi(\Sigma^{l-1})(x, x') = \Phi^l(1)\hat{\phi}\left(\frac{\Phi^{l-1}(c)}{\Phi^{l-1}(1)}\right). \tag{18}$$

Substituting the Taylor expansion of $\frac{\Phi^{l-1}(c)}{\Phi^{l-1}(1)}$ into the Taylor expansion of $\hat{\phi}$ given by Lem G.4 gives us a Taylor series whose coefficients are all nonnegative, since those of $\frac{\Phi^{l-1}(c)}{\Phi^{l-1}(1)}$ and $\hat{\phi}$ are nonnegative as well. In addition, plugging in 1 for c in this Taylor series shows that the sum of the coefficients equal $\hat{\phi}(1) = 1$, so the series is absolutely convergent for $c \in [-1, 1]$. This proves the inductive step.

For the NTK, the proof is similar, except we now also have a product step where we multiply $\mathrm{V}_{\phi'}(\Sigma^{l-1})$ with Θ^{l-1}, and we simply just need to use the fact that product of two Taylor series with nonnegative coefficients is another Taylor series with nonnegative coefficients, and that the resulting radius of convergence is the minimum of the original two radii of convergence.

□

Lemma G.4. *Consider any $\sigma > 0$ and any $\phi : \mathbb{R} \to \mathbb{R}$ square-integrable against $\mathcal{N}(0, \sigma^2)$. Then the function*

$$\hat{\phi} : [-1, 1] \to [-1, 1], \quad \hat{\phi}(c) \stackrel{\text{def}}{=} \frac{\mathbb{E}\,\phi(x)\phi(y)}{\mathbb{E}\,\phi(x)^2}, \quad \textit{where } (x, y) \sim \mathcal{N}\left(0, \begin{pmatrix} \sigma^2 & \sigma^2 c \\ \sigma^2 c & \sigma^2 \end{pmatrix}\right),$$

has a Taylor series expansion around 0

$$\hat{\phi}(c) = a_0 + a_1 c + a_2 c^2 + \cdots \tag{19}$$

with the properties that $a_i \geq 0$ for all i and $\sum_i a_i = \hat{\phi}(1) = 1$, so that Eq. (19) is absolutely convergent on $c \in [-1, 1]$.

Proof. Since the denominator of $\hat{\phi}$ doesn't depend on c, it suffices to prove this Taylor expansion exists for its numerator. Let $\tilde{\phi}(c) \stackrel{\text{def}}{=} \phi(\sigma c)$. Then

$$\mathbb{E}\left[\phi(x)\phi(y) : (x, y) \sim \mathcal{N}\left(0, \begin{pmatrix} \sigma^2 & \sigma^2 c \\ \sigma^2 c & \sigma^2 \end{pmatrix}\right)\right] = \mathbb{E}\left[\tilde{\phi}(x)\tilde{\phi}(y) : (x, y) \sim \mathcal{N}\left(0, \begin{pmatrix} 1 & c \\ c & 1 \end{pmatrix}\right)\right]$$

By Fact G.5, this is just

$$b_0^2 + b_1^2 c + b_2^2 c^2 + \cdots$$

where b_i are the Hermite coefficients of $\tilde{\phi}$. Since $\phi \in L^2(\mathcal{N}(0, \sigma^2))$,

$$\sum_i b_i^2 < \infty$$

as desired. □

Fact G.5 (O'Donnell (2014)). *For any $\phi, \psi \in L^2(\mathcal{N}(0,1))$,*

$$\mathbb{E}\left[\phi(x)\psi(y) : (x,y) \sim \mathcal{N}\left(0, \begin{pmatrix} 1 & c \\ c & 1 \end{pmatrix}\right)\right] = a_0 b_0 + a_1 b_1 c + a_2 b_2 c^2 + \cdots$$

where a_i and b_i are respectively the Hermite coefficients of ϕ and ψ.

Fixing k, taking the $d \to \infty$ limit In this subsection, we prove that $\mu_k \sim d^{-k}$ as $d \to \infty$ with k fixed. More precisely,

Theorem G.6. *Let K be the CK or NTK of an MLP on a boolean cube $\square^d$. Then K can be expressed as $K(x,y) = \Phi(\langle x, y\rangle/d)$ for some $\Phi : \mathbb{R} \to \mathbb{R}$. If we fix k and let $d \to \infty$, then*

$$\lim_{d\to\infty} d^k \mu_k = \Phi^{(k)}(0),$$

where $\Phi^{(k)}$ denotes the kth derivative of Φ.

An immediate consequence, using the fact that the degree k eigenspace has dimension $\binom{d}{k} \sim d^k/k!$ for large d, is

Theorem 5.1 (Asymptotic Fractional Variance). *Let K be the CK or NTK of an MLP on a boolean cube $\square^d$. Then K can be expressed as $K(x,y) = \Phi(\langle x, y\rangle/d)$ for some analytic function $\Phi : \mathbb{R} \to \mathbb{R}$. If we fix k and let the input dimension $d \to \infty$, then the fractional variance of degree k converges to*

$$(k!)^{-1}\Phi^{(k)}(0)/\Phi(1) = \frac{(k!)^{-1}\Phi^{(k)}(0)}{\sum_{j\geq 0}(j!)^{-1}\Phi^{(j)}(0)}$$

where $\Phi^{(k)}$ denotes the kth derivative of Φ.

Let's first give some intuition for why Thm G.6 should be true. By Eqs. (8) and (5),

$$\begin{aligned}\mu_k &= \left(\frac{1+\mathcal{T}_\Delta}{2}\right)^{d-k}\left(\frac{1-\mathcal{T}_\Delta}{2}\right)^{k}\Phi(1) \\ &= \frac{1}{2^d}\sum_{r=0}^{d-k}\binom{d-k}{r}\Phi^{[k]}(1-r\Delta)\end{aligned}$$

where $\Phi^{[k]} = (1-\mathcal{T}_\Delta)^k\,\Phi$ is the kth backward finite difference with step $\Delta = 2/d$. Approximating the finite difference with derivative, we thus would expect that, as suggested by Lem G.8,

$$\Phi^{[k]}(x) \approx \Delta^k \Phi^{(k)}(x) = (2/d)^k \Phi^{(k)}(x), \quad \text{when } d \text{ is large,}$$

where $\Phi^{(k)}$ is the kth derivative of Φ. Since $\frac{1}{2^{d-k}}\binom{d-k}{r}$ is the probability mass at $1 - r\Delta$ of the binomial variable $B \stackrel{\text{def}}{=} \frac{1}{d-k}\sum_{i=1}^{d-k} X_i$, where X_i is the Bernoulli variable taking ± 1 value with half chance each. This random variable converges almost surely to the delta distribution at 0, by law of large numbers. Therefore, we would expect

$$\mu_k \sim (1/d)^k \Phi^{(k)}(0)$$

as $d \to \infty$.

There is a single difficulty when trying to formalize this intuition. One is the possibility that the kth derivative $\Phi^{(k)}$ does not exist at the endpoints 1 and -1 (note that it must exist in the interior, because by Thm G.3, Φ is analytic on $(-1,1)$). In this case, we must show that the portion of the interval $[-1,1]$ near the endpoints contributes exponentially little mass to the probability distribution of the binomial variable B, that it suppresses any blowup that can happen at the edge.

We formalize this reasoning in the proof below.

Proof. We first prevent any blowup that may happen at the edge. Let $\Phi^{[k]}$ be the kth backward difference of Φ with step size $\Delta = 2/d$, as in Lem G.8. First note the trivial bound

$$\begin{aligned}\Phi^{[k]}(x) &= \sum_{s=0}^{k}(-1)^s\binom{k}{s}\Phi(x-s\Delta) \leq \sum_{s=0}^{k}\binom{k}{s}\Phi(x-s\Delta) \\ &\leq 2^k\Phi(x) \leq 2^k\Phi(1)\end{aligned}$$

since Φ is nondecreasing by Lem G.7. Then for any $R \in [0, (d-k)/2 - 1]$,

$$\begin{aligned} d^k \mu_k &= \frac{d^k}{2^d} \sum_{r=0}^{d-k} \binom{d-k}{r} \Phi^{[k]}(1 - r\Delta) \\ &= \frac{d^k}{2^d} \sum_{r=R+1}^{d-k-R-1} \binom{d-k}{r} \Phi^{[k]}(1 - r\Delta) + \frac{d^k}{2^d} \left(\sum_{r=0}^{R} + \sum_{r=d-k-R}^{d-k} \right) \binom{d-k}{r} \Phi^{[k]}(1 - r\Delta). \end{aligned}$$

Now, in the second term, $\Phi^{[k]}(1 - r\Delta) \leq 2^k \Phi(1)$ as above, so that, by the binomial coefficient entropy bound (Fact G.9),

$$\begin{aligned} 0 \leq d^k \mu_k - \frac{d^k}{2^d} \sum_{r=R+1}^{d-k-R-1} \binom{d-k}{r} \Phi^{[k]}(1 - r\Delta) &\leq \frac{d^k}{2^d} \left(\sum_{r=0}^{R} + \sum_{r=d-k-R}^{d-k} \right) \binom{d-k}{r} 2^k \Phi(1) \\ &\leq \frac{d^k}{2^{d-k}} 2 \cdot 2^{H(p)(d-k)} \Phi(1) \\ &= \frac{d^k \Phi(1)}{2^{(1-H(p))d - (1-H(p))k - 1}}, \end{aligned} \tag{20}$$

where $p = R/(d-k)$. If we choose $R = \lfloor \tilde{p}(d-k) \rfloor$ for a fixed $\tilde{p} < 1/10$, then $H(p) < 1$, and (20) is decreasing exponentially fast to 0 with d.

Now we formalize the law of large number intuition. It then suffices to show that

$$\frac{d^k}{2^d} \sum_{r=R+1}^{d-k-R-1} \binom{d-k}{r} \Phi^{[k]}(1 - r\Delta) \to \Phi^{(k)}(0).$$

We will do so by presenting an upper and a lower bound which both converge to this limit.

We first discuss the upper bound. By Lem G.8,

$$\frac{d^k}{2^d} \sum_{r=R+1}^{d-k-R-1} \binom{d-k}{r} \Phi^{[k]}(1 - r\Delta) \leq \frac{1}{2^{d-k}} \sum_{r=R+1}^{d-k-R-1} \binom{d-k}{r} \Phi^{(k)}(1 - r\Delta).$$

The RHS can be upper bounded by an expectation over a binomial random variable: Let X_i be ± 1 Bernoulli variables, and let $\Phi^{(k)}_{\frac{1}{2}\tilde{p}}$ be the function

$$\Phi^{(k)}_{\frac{1}{2}\tilde{p}}(x) = \begin{cases} \Phi^{(k)}(x) & \text{if } x \in [-1 + \frac{1}{2}\tilde{p}, 1 - \frac{1}{2}\tilde{p}] \\ 0 & \text{otherwise} \end{cases}$$

Note that $\Phi^{(k)}_{\frac{1}{2}\tilde{p}}(x)$ is a bounded function. Then with the binomial variable $B = \frac{1}{d} \sum_{i=1}^{d-k} (X_i + k/(d-k))$, we have

$$\frac{1}{2^{d-k}} \sum_{r=R+1}^{d-k-R-1} \binom{d-k}{r} \Phi^{(k)}(1 - r\Delta) \leq \mathbb{E}\, \Phi^{(k)}_{\frac{1}{2}\tilde{p}}(B).$$

By strong law of large numbers, B converges almost surely to 0 as $d \to \infty$ with k fixed. Thus the RHS converges to

$$\Phi^{(k)}_{\frac{1}{2}\tilde{p}}(0) = \Phi^{(k)}(0)$$

as desired.

A similar argument proceeds for the lower bound, after noting that, by Lem G.8,

$$\frac{1}{2^{d-k}} \sum_{r=R+1}^{d-k-R-1} \binom{d-k}{r} \Phi^{(k)}(1 - (r+k)\Delta) \leq \frac{d^k}{2^d} \sum_{r=R+1}^{d-k-R-1} \binom{d-k}{r} \Phi^{[k]}(1 - r\Delta).$$

□

Lemma G.7. *Let K be the CK or NTK of an MLP with domain $\sqrt{d}\mathcal{S}^{d-1} \subseteq \mathbb{R}^d$. Then $K(x, y) = \Phi\left(\frac{\langle x,y\rangle}{\|x\|\|y\|}\right)$ where $\Phi : [-1, 1] \to \mathbb{R}$ is analytic on $(-1, 1)$, and all derivatives of Φ are nonnegative on $(-1, 1)$.*

Proof. Note that, by Thm G.3, Φ has a convergent Taylor expansion on $(-1, 1)$, i.e Φ *is analytic on* $(-1, 1)$. Thus, all derivatives of Φ exist (and are finite) on the interval $(-1, 1)$ and have the obvious Taylor series derived from Eq. (17). For example,

$$\Phi'(c) = a_1 c + 2a_2 c + 3a_3 c^2 + \cdots .$$

Such Taylor series also converge on the open interval $(-1, 1)$ (but could diverge on the endpoints -1 and 1). We get the desired result by noting that all a_i are nonnegative. □

By expressing finite difference as an integral, it's easy to see that

Lemma G.8. *With the same setting as in Lem G.7, let $\Phi^{[k]}$ be the kth backward finite difference with step size $\Delta = 2/d$,*

$$\Phi^{[k]}(x) \stackrel{\text{def}}{=} (1 - \mathcal{T}_\Delta)^k \Phi(x).$$

Then $\Phi^{[k]}$ is analytic on $(-1, 1)$, has all derivatives nonnegative, and

$$0 \le \Delta^k \Phi^{(k)}(x - k\Delta) \le \Phi^{[k]}(x) \le \Delta^k \Phi^{(k)}(x), \tag{21}$$

where $\Phi^{(k)}$ is the kth derivative of Φ.

Proof. Everything follows immediately from

$$\Phi^{[k]}(x) = \int_0^\Delta \cdots \int_0^\Delta \Phi^{(k)}(x - h_1 - \cdots - h_k)\, \mathrm{d}h_1 \cdots \mathrm{d}h_k$$

and Lem G.7. □

Fact G.9 (Entropy bound on sum of binomial coefficients)**.** *For any $k \le d$ and $R \le d/2$,*

$$\sum_{r=0}^{R} \binom{d-k}{r} \le 2^{H(p)(d-k)}$$

where $p = R/(d - k)$, and H is the binary entropy

$$H(p) = -p \log_2 p - (1 - p) \log_2(1 - p).$$

H THE $\{0, 1\}^d$ VS THE $\{\pm 1\}^d$ BOOLEAN CUBE

Valle-Pérez et al. (2018) actually did their experiments on the $\{0, 1\}^d$ boolean cube, whereas here, we have focused on the $\{\pm 1\}^d$ boolean cube. As datasets are typically centered before feeding into a neural network (for example, using Pytorch's `torchvision.transform.Normalize`), $\{\pm 1\}^d$ is much more natural. In comparison, using the $\{0, 1\}^d$ cube is equivalent to adding a bias in the input of a network and reducing the weight variance in the input layer, since any $x \in \{\pm 1\}^d$ corresponds to $\frac{1}{2}(x + 1) \in \{0, 1\}^d$. As such, one would expect there is *more bias toward low frequency components with inputs from* $\{0, 1\}^d$.

Nevertheless, here we verify that our observations of Section 4 above still holds over the $\{0, 1\}^d$ cube by repeating the same experiments as Fig. 1 in this setting (Fig. 14). Just like over the $\{\pm 1\}^d$ cube, the relu network biases significantly toward certain functions, but with erf, and with increasing σ_w^2, this lessens. With depth 32 and σ_w^2, the boolean functions obtained from erf network see no bias at all.

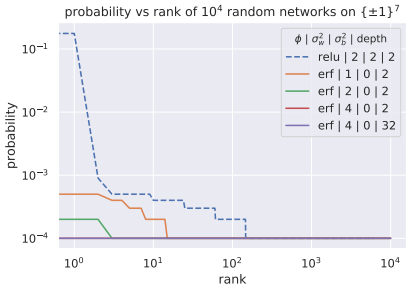


Figure 14: **The same experiments as Fig. 1 but over $\{0, 1\}^7$.**