Learning sparse features can lead to overfitting in neural networks

Leonardo Petrini *
Institute of Physics
École Polytechnique Fédérale de Lausanne
leonardo.petrini@epfl.ch

Francesco Cagnetta *
Institute of Physics
École Polytechnique Fédérale de Lausanne
francesco.cagnetta@epfl.ch

Eric Vanden-Eijnden
Courant Institute of Mathematical Sciences
New York University
eve2@cims.nyu.edu

Matthieu Wyart
Institute of Physics
École Polytechnique Fédérale de Lausanne
matthieu.wyart@epfl.ch

Abstract

It is widely believed that the success of deep networks lies in their ability to learn a meaningful representation of the features of the data. Yet, understanding when and how this feature learning improves performance remains a challenge: for example, it is beneficial for modern architectures trained to classify images, whereas it is detrimental for fully-connected networks trained on the same data. Here we propose an explanation for this puzzle, by showing that feature learning can perform worse than lazy training (via random feature kernel or the NTK) as the former can lead to a sparser neural representation. Although sparsity is known to be essential for learning anisotropic data, it is detrimental when the target function is constant or smooth along certain directions of input space. We illustrate this phenomenon in two settings: *(i)* regression of Gaussian random functions on the d-dimensional unit sphere and *(ii)* classification of benchmark datasets of images. For *(i)*, we compute the scaling of the generalization error with the number of training points and show that methods that do not learn features generalize better, even when the dimension of the input space is large. For *(ii)*, we show empirically that learning features can indeed lead to sparse and thereby less smooth representations of the image predictors. This fact is plausibly responsible for deteriorating the performance, which is known to be correlated with smoothness along diffeomorphisms.

1 Introduction

Neural networks are responsible for a technological revolution in a variety of machine learning tasks. Many such tasks require learning functions of high-dimensional inputs from a finite set of examples, thus should be generically hard due to the *curse of dimensionality* [1, 2]: the exponent that controls the scaling of the generalization error with the number of training examples is inversely proportional to the input dimension d. For instance, for standard image classification tasks with d ranging in $10^3 \div 10^5$, such exponent should be practically vanishing, contrary to what is observed in practice [3]. In this respect, understanding the success of neural networks is still an open question. A popular explanation is that, during training, neurons adapt to features in the data that are relevant for the task [4], effectively reducing the input dimension and making the problem tractable [5–7]. However, understanding quantitatively if this intuition is true and how it depends on the structure of the task remains a challenge.

[*] Equal contribution (a coin was flipped).

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

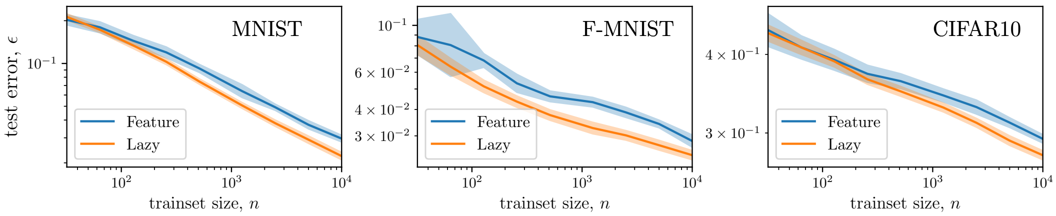


Figure 1: **Feature vs. Lazy in image classification.** Generalization error as a function of the training-set size n for infinite-width fully-connected networks (FCNs) trained in the feature (blue) and lazy regime (orange). In the latter case the limit is taken exactly by training an SVC algorithm with the analytical NTK [23]. In the former case, the infinite-width limit can be accurately approximated for these datasets by considering very wide nets ($H = 10^3$), and performing ensemble averaging on different initial conditions of the parameters as shown in [24, 25]. Panels correspond to different benchmark image datasets [26–28]. Results are averaged over 10 different initializations of the networks and datasets.

Recently much progress was made in characterizing the conditions which lead to features learning, in the overparameterized setting where networks generally perform best. When the initialization scale of the network parameters is large [8] one encounters the *lazy training regime*, where neural networks behave as kernel methods [9, 10] (coined Neural Tangent Kernel or NTK) and features are not learned. By contrast, when the initialization scale is small, a *feature learning regime* is found [11–13] where the network parameters evolve significantly during training. This limit is much less understood apart from very simple architectures, where it can be shown to lead to sparse representations where a limited number of neurons are active after training [14]. Such sparse representations can also be obtained by regularizing the weights during training [2, 15].

In terms of performance, most theoretical works have focused on fully-connected networks. For these architectures, feature learning was shown to significantly outperform lazy training [16–19, 11] for certain tasks, including approximating a function which depends only on a subset or a linear combination of the input variables. However, when such primitive networks are trained on image datasets, learning features is detrimental [20, 21], as illustrated in Fig. 1 (see [19, Fig. 3] for the analogous plot in the case of a target function depending on just one of the input variables, where learning features is beneficial). A similar result was observed in simple models of data [22]. These facts are unexplained, yet central to understanding the implicit bias of the feature learning regime.

1.1 Our contribution

Our main contribution is to provide an account of the drawbacks of learning sparse representations based on the following set of ideas. Consider, for concreteness, an image classification problem: *(i)* images class varies little along smooth deformations of the image; *(ii)* due to that, tasks like image classification require a continuous distribution of neurons to be represented; *(iii)* thus, requiring sparsity can be detrimental for performance. We build our argument as follows.

- In order to find a quantitative description of the phenomenon, we start from the problem of regression of a random target function of controlled smoothness on the d-dimensional unit sphere, and study the property of the minimizers of the empirical loss with n observations, both in the lazy and the feature learning regimes. More specifically, we consider two extreme limits—the NTK limit and mean-field limit—as representatives of lazy and feature regimes, respectively (section 2). Both these limits admit a simple formulation that allows us to predict generalization performances. In particular, our results on feature learning rely on solutions having an atomic support. This property can be justified for one-hidden-layer neural networks with ReLU activations and weight decay. Yet, we also find such a sparsity empirically using gradient descent in the absence of regularization, if weights are initialized to be small enough.
- We find that lazy training leads to smoother predictors than feature learning. As a result, lazy training outperforms feature learning when the target function is also sufficiently smooth. Otherwise, the performances of the two methods are comparable, in the sense that they display the same asymptotic decay of generalization error with the number of training

examples. Our predictions are obtained from asymptotic arguments that we systematically back up with numerical studies.

- For image datasets, it is believed that diffeomorphisms of images are key transformations along which the predictor function should only mildly vary to obtain good performance [29]. From the results above, a natural explanation as to why lazy beats feature for fully connected networks is that it leads to predictors with smaller variations along diffeomorphisms. We confirm that this is indeed the case empirically on benchmark datasets.

Numerical experiments are performed in PyTorch [30], and the code for reproducing experiments is available online at github.com/pcsl-epfl/regressionsphere.

1.2 Related Work

The property that training ReLU networks in the feature regime leads to a sparse representation was observed empirically [31]. This property can be justified for one-hidden-layer networks by casting training as a L1 minimization problem [32, 2], then using a representer theorem [33, 15, 34]. This is analogous to what is commonly done in predictive sparse coding [35–38].

Many works have investigated the benefits of learning sparse representations in neural networks. [2, 16–19, 39, 40] study cases in which the true function only depends on a linear subspace of input space, and show that feature learning profitably capture such property. Even for more general problems, sparse representations of the data might emerge naturally during deep network training—a phenomenon coined *neural collapse* [41]. Similar sparsification phenomena, for instance, have been found to allow for learning convolutional layers from scratch [42, 43]. Our work builds on this body of literature by pointing out that learning sparse features can be detrimental, if the task does not allow for it.

There is currently no general framework to predict rigorously the learning curve exponent β defined as $\epsilon(n) = \mathcal{O}(n^{-\beta})$ for kernels. Some of our asymptotic arguments can be obtained by other approximations, such as assuming that data points lie on a lattice in $\mathbb{R}^d$ [44], or by using the non-rigorous replica method of statistical physics [45–47]. In the case $d = 2$, we provide a more explicit mathematical formulation of our results, which leads to analytical results for certain kernels. We systematically back up our predictions with numerical tests as d varies.

Finally, in the context of image classification, the connection between performance and 'stability' or smoothness toward small diffeomorphisms of the inputs has been conjectured by [29, 48]. Empirically, a strong correlation between these two quantities was shown to hold across various architectures for real datasets [49]. In that reference, it was found that fully connected networks lose their stability over training: here we show that this effect is much less pronounced in the lazy regime.

2 Problem and notation

Task We consider a supervised learning scenario with n training points $\{\boldsymbol{x}_i\}_{i=1}^n$ uniformly drawn on the d-dimensional unit sphere $\mathbb{S}^{d-1}$. We assume that the target function f^* is an isotropic Gaussian random process on $\mathbb{S}^{d-1}$ and control its statistics via the spectrum: by introducing the decomposition of f^* into spherical harmonics (see App. A for definitions),

$$f^*(\boldsymbol{x}) = \sum_{k\geq 0} \sum_{\ell=1}^{\mathcal{N}_{k,d}} f^*_{k,\ell} Y_{k,\ell}(\boldsymbol{x}) \quad \text{with} \quad \mathbb{E}\left[f^*_{k,\ell}\right] = 0, \quad \mathbb{E}\left[f^*_{k,\ell} f^*_{k',\ell'}\right] = c_k \delta_{k,k'} \delta_{\ell,\ell'}. \tag{2.1}$$

We assume that all the c_k with k odd vanish apart from c_1: this is required to guarantee that f^* can be approximated as well as desired with a one-hidden-layer ReLU network with no biases, as discussed in App. A. We also assume that the non-zero c_k decay as a power of k for $k \gg 1$, $c_k \sim k^{-2\nu_t-(d-1)}$. The exponent $\nu_t > 0$ controls the (weak) differentiability of f^* on the sphere (see App. A) and also the statistics of f^* in real space:

$$\mathbb{E}\left[|f^*(\boldsymbol{x}) - f^*(\boldsymbol{y})|^2\right] = O\left(|\boldsymbol{x} - \boldsymbol{y}|^{2\nu_t}\right) = O\left((1 - \boldsymbol{x}\cdot\boldsymbol{y})^{\nu_t}\right) \quad \text{as} \quad \boldsymbol{x} \to \boldsymbol{y}. \tag{2.2}$$

Examples of such a target function for $d = 3$ and different values of ν_t are reported in Fig. 2.

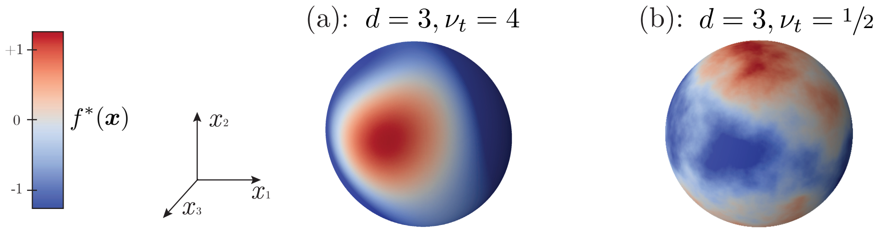


Figure 2: **Gaussian random process on the sphere.** We show here two samples of the task introduced in section 2 when the target function $f^*(\boldsymbol{x})$ is defined on the 3−dimensional unit sphere. (a) and (b) show samples of large and small smoothness coefficient ν_t, respectively.

Neural network representation in the feature regime In this regime we aim to approximate the target function $f^*(x)$ via a *one-hidden-layer neural network* of width H,

$$f_H(\boldsymbol{x}) = \frac{1}{H}\sum_{h=1}^{H} w_h \sigma(\boldsymbol{\theta}_h \cdot \boldsymbol{x}), \tag{2.3}$$

where $\{\boldsymbol{\theta}_h\}_{h=1}^H$ (the features) and $\{w_h\}_{h=1}^H$ (the weights) are the network parameters to be optimized, and $\sigma(x)$ denotes the ReLU function, $\sigma(x)=\max\{0,x\}$. If we assume that $\{\boldsymbol{\theta}_h, w_h\}_{h=1}^H$ are independently drawn from a probability measure μ on $\mathbb{S}^{d-1}\times\mathbb{R}$ such that the Radon measure $\gamma=\int_{\mathbb{R}} w\mu(\cdot, dw)$ exists, then as $H\to\infty$,

$$\lim_{H\to\infty} f_H(\boldsymbol{x}) = \int_{\mathbb{S}^{d-1}} \sigma(\boldsymbol{\theta}\cdot\boldsymbol{x})d\gamma(\boldsymbol{\theta}) \qquad \text{a.e. on } \mathbb{S}^{d-1}. \tag{2.4}$$

This is the so-called mean-field limit [11, 12], and it is then natural to determine the optimal γ via

$$\gamma^* = \arg\min_{\gamma} \int_{\mathbb{S}^{d-1}} |d\gamma(\boldsymbol{\theta})| \quad \text{subject to:} \quad \int_{\mathbb{S}^{d-1}} \sigma(\boldsymbol{\theta}\cdot\boldsymbol{x}_i)d\gamma(\boldsymbol{\theta}) = f^*(\boldsymbol{x}_i) \quad \forall i=1,\dots,n. \tag{2.5}$$

In practice, we can approximate this minimization problem by using a network with large but finite width, constraining the feature to be on the sphere $|\boldsymbol{\theta}_h|=1$, and minimizing the following empirical loss with L1 regularization on the weights,

$$\min_{\substack{\{w_h,\theta_h\}_{h=1}^H \\ |\boldsymbol{\theta}_h|=1}} \frac{1}{2n}\sum_{i=1}^{n}\left(f^*(\boldsymbol{x}_i) - \frac{1}{H}\sum_{h=1}^{H} w_h\sigma(\boldsymbol{\theta}_h\cdot\boldsymbol{x}_i)\right)^2 + \frac{\lambda}{H}\sum_{h=1}^{H}|w_h|. \tag{2.6}$$

This minimization problem leads to (2.5) when $H\to\infty$ and $\lambda\to 0$. Note that, by homogeneity of ReLU, (2.6) can be shown to be equivalent to imposing a regularization on the L2 norm of all parameters [32, Thm. 10], i.e. the usual weight decay.

To proceed we will make the following assumption about the minimizer γ^*:

Assumption 1. *The minimizer γ^* of (2.5) is unique and atomic, with $n_A\le n$ atoms, i.e. there exists $\{w_i^*, \boldsymbol{\theta}_i^*\}_{i=1}^{n_A}$ such that*

$$\gamma^* = \sum_{i=1}^{n_A} w_i^* \delta_{\boldsymbol{\theta}_i^*}. \tag{2.7}$$

The main component of the assumption is the uniqueness of γ^*; if it holds the sparsity of γ^* follows from the representer theorem, see e.g. [33]. Both the uniqueness and sparsity of the minimizer can be justified as holding generically using asymptotic arguments involving recasting the $L1$ minimization problem 2.5 as a linear programming one: these arguments are standard (see e.g. [50]) and are presented in App. B for the reader convenience. In our arguments below to deduce the scaling of the generalization error we will mainly use that $n_A = O(n)$—we shall confirm this fact numerically even in the absence of regularization, if the weights are initialized to be small enough. Notice that from Assumption 1 it follows that the predictor in the feature regime corresponding to the minimizer γ^* takes the following form

$$f^{\text{FEATURE}}(\boldsymbol{x}) = \sum_{i=1}^{n_A} w_i^* \sigma(\boldsymbol{\theta}_i^*\cdot\boldsymbol{x}). \tag{2.8}$$

Neural network representation in the lazy regime. In this regime we approximate the target function $f^*(x)$ via

$$f^{\text{NTK}}(\boldsymbol{x}) = \sum_{i=1}^{n} g_i K^{\text{NTK}}(\boldsymbol{x}_i \cdot \boldsymbol{x}), \tag{2.9}$$

where the weights $\{g_i\}_{i=1}^n$ solve

$$f^*(\boldsymbol{x}_j) = \sum_{i=1}^{n} g_i K^{\text{NTK}}(\boldsymbol{x}_i \cdot \boldsymbol{x}_j), \qquad j = 1, \dots, n. \tag{2.10}$$

and $K^{\text{NTK}}(\boldsymbol{x} \cdot \boldsymbol{y})$ is the *Neural Tangent Kernel* (NTK) [9]

$$K^{\text{NTK}}(\boldsymbol{x} \cdot \boldsymbol{y}) = \int_{\mathbb{S}^{d-1} \times \mathbb{R}} \left(\sigma(\boldsymbol{\theta} \cdot \boldsymbol{x})\sigma(\boldsymbol{\theta} \cdot \boldsymbol{y}) + w^2 \, \boldsymbol{x} \cdot \boldsymbol{y} \, \sigma'(\boldsymbol{\theta} \cdot \boldsymbol{x})\sigma'(\boldsymbol{\theta} \cdot \boldsymbol{y}) \right) d\mu_0(\boldsymbol{\theta}, w). \tag{2.11}$$

Here μ_0 is a fixed probability distribution which, in the NTK training regime [9], is the distribution of the features and weights at initialization. It is well-known [51] that the solution to kernel ridge regression problem can also be expressed via the kernel trick as

$$f^{\text{NTK}}(\boldsymbol{x}) = \int_{\mathbb{S}^{d-1} \times \mathbb{R}} \left(g_w(\boldsymbol{\theta}, w)\sigma(\boldsymbol{\theta} \cdot \boldsymbol{x}) + w\boldsymbol{x} \cdot \boldsymbol{g}_{\boldsymbol{\theta}}(\boldsymbol{\theta}, w)\sigma'(\boldsymbol{\theta} \cdot \boldsymbol{x}) \right) d\mu_0(\boldsymbol{\theta}, w) \tag{2.12}$$

where $\boldsymbol{g}_{\boldsymbol{\theta}}$ and g_w are the solutions of

$$\begin{aligned} &\min_{g_w, \boldsymbol{g}_{\boldsymbol{\theta}}} \int_{\mathbb{S}^{d-1} \times \mathbb{R}} \left(g_w^2(w, \boldsymbol{\theta}) + |\boldsymbol{g}_{\boldsymbol{\theta}}(w, \boldsymbol{\theta})|^2 \right) d\mu_0(\boldsymbol{\theta}, w) \\ &\text{subject to: } \int_{\mathbb{S}^{d-1} \times \mathbb{R}} \left(g_w(w, \boldsymbol{\theta})\sigma(\boldsymbol{\theta} \cdot \boldsymbol{x}_i) + w\boldsymbol{x}_i \cdot \boldsymbol{g}_{\boldsymbol{\theta}}(w, \boldsymbol{\theta})\sigma'(\boldsymbol{\theta} \cdot \boldsymbol{x}_i) \right) d\mu_0(\boldsymbol{\theta}, w) = f^*(\boldsymbol{x}_i) \\ &\qquad \forall i = 1, \dots, n. \end{aligned} \tag{2.13}$$

Another lazy limit can be obtained equivalently by training only the weights while keeping the features to their initialization value. This is equivalent to forcing $\boldsymbol{g}_{\boldsymbol{\theta}}(\boldsymbol{\theta}, w)$ to vanish in Eq. 2.13, resulting again in a kernel method. The kernel, in this case, is called *Random Feature Kernel* (K^{RFK}), and can be obtained from Eq. 2.11 by setting $d\mu_0(\boldsymbol{\theta}, w) = \delta_{w=0} d\tilde{\mu}_0(\boldsymbol{\theta})$. The minimizer can then be written as in Eq. 2.9 with K^{NTK} replaced by K^{RFK}.

3 Asymptotic analysis of generalization

In this section, we characterize the asymptotic decay of the generalization error $\bar{\epsilon}(n)$ averaged over several realizations of the target function f^*. Denoting with $d\tau^{d-1}(\boldsymbol{x})$ the uniform measure on $\mathbb{S}^{d-1}$,

$$\bar{\epsilon}(n) = \mathbb{E}_{f^*} \left[\int d\tau^{d-1}(\boldsymbol{x}) \, \left(f^n(\boldsymbol{x}) - f^*(\boldsymbol{x}) \right)^2 \right] = \mathcal{A}_d n^{-\beta} + o(n^{-\beta}), \tag{3.1}$$

for some constant $\mathcal{A}_d$ which might depend on d but not on n. Both for the lazy (see Eq. 2.9) and feature regimes (see Eq. 2.8) the predictor can be written as a sum of $\mathcal{O}(n)$ terms:

$$f^n(\boldsymbol{x}) = \sum_{j=1}^{\mathcal{O}(n)} g_j \varphi(\boldsymbol{x} \cdot \boldsymbol{y}_j) := \int_{\mathbb{S}^{d-1}} g^n(\boldsymbol{y})\varphi(\boldsymbol{x} \cdot \boldsymbol{y}) d\tau(\boldsymbol{y}). \tag{3.2}$$

In the feature regime, the g_j's ($\boldsymbol{y}_j$) coincide with the optimal weights w_j^* (features $\boldsymbol{\theta}_j^*$), φ with the activation function σ. In the lazy regime, the $\boldsymbol{y}_j$ are the training points $\boldsymbol{x}_j$, φ is the neural tangent or random feature kernel the g_j's are the weights solving Eq. 2. We have defined the density $g^n(\boldsymbol{x}) = \sum_j |\mathbb{S}^{d-1}| g_j \delta(\boldsymbol{x} - \boldsymbol{y}_j)$ so as to cast the predictor as a convolution on the sphere. Therefore, the projections of f^n onto spherical harmonics $Y_{k,\ell}$ read $f_{k,\ell}^n = g_{k,\ell}^n \varphi_k$, where $g_{k,\ell}^n$ is the projection of $g^n(\boldsymbol{x})$ and φ_k that of $\varphi(\boldsymbol{x} \cdot \boldsymbol{y})$. For ReLU neurons one has (as shown in App. A)

$$\varphi_k^{\text{LAZY}} \sim k^{-(d-1)-2\nu} \quad \text{with } \nu = 1/2 \text{ (NTK)}, 3/2 \text{ (RFK)}, \quad \varphi_k^{\text{FEATURE}} \sim k^{-\frac{d-1}{2}-3/2}. \tag{3.3}$$

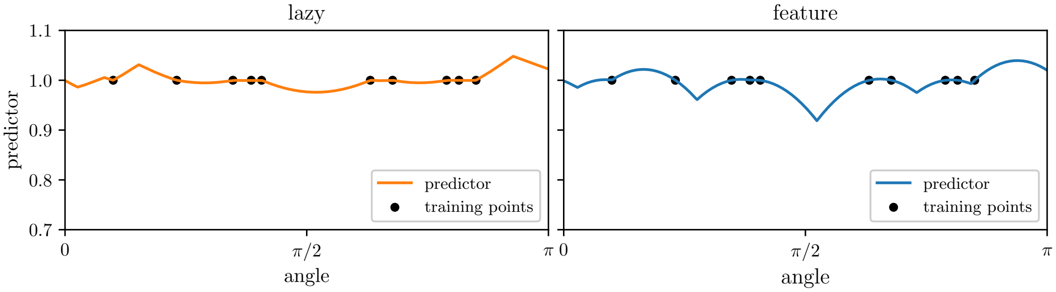


Figure 3: **Feature vs. Lazy Predictor.** Predictor of the lazy (left) and feature (right) regime when learning the constant function on the ring with 8 uniformly-sampled training points.

Main Result Consider a target function f^* with smoothness exponent ν_t as defined above, with data lying on $\mathbb{S}^{d-1}$. If f^* is learnt with a one-hidden-layer network with ReLU neurons in the regimes specified above, then the generalization error follows $\bar{\epsilon}(n) \sim n^{-\beta}$ with:

$$\beta^{\text{LAZY}} = \frac{\min\{2(d-1)+4\nu, 2\nu_t\}}{d-1} \quad \text{with} \quad \nu = \begin{cases} 1/2 \text{ for NTK}, \\ 3/2 \text{ for RFK}, \end{cases}, \tag{3.4a}$$

$$\beta^{\text{FEATURE}} = \frac{\min\{(d-1)+3, 2\nu_t\}}{d-1}. \tag{3.4b}$$

This is our central result. It implies that if the target function is a smooth isotropic Gaussian field (realized for large ν_t), then lazy beats feature, in the sense that training the network in the lazy regime leads to a better scaling of the generalization performance with the number of training points.

Strategy There is no general framework for a rigorous derivation of the generalization error in the ridgeless limit $\lambda \to 0$: predictions such as that of Eq. 3.4 can be obtained by either assuming that training points (for Eq. 3.4a) and neurons (for Eq. 3.4b) lie on a periodic lattice [44], or (for Eq. 3.4a) using the replica method from physics [45] as shown in App. F. Here we follow a different route, by first characterizing the form of the predictor for $d=2$ (proof in App. C). This property alone allows us to determine the asymptotic scaling of the generalization error. We use it to analytically obtain the generalization error in the NTK case with a slightly simplified function φ (details in App. D). This calculation motivates a simple ansatz for the form of $g^n(\boldsymbol{x})$ entering Eq. 3.2 and its projections onto spherical harmonics, which extends naturally to arbitrary dimension. We confirm the predictions resulting from this ansatz systematically in numerical experiments.

Properties of the predictor in $d=2$ On the unit circle $\mathbb{S}^1$ all points are identified by a polar angle $x \in [0, 2\pi)$. Hence both target function and estimated predictor are functions of the angle, and all functions of a scalar product are in fact functions of the difference in angle. In particular, introducing $\tilde{\varphi}(x) = \varphi(\cos(x))$,

$$f^n(x) = \sum_j g_j \tilde{\varphi}(x - x_j) \equiv \int_0^{2\pi} \frac{dy}{2\pi} g^n(y) \tilde{\varphi}(x-y), \tag{3.5}$$

where we defined

$$g^n(x) = \sum_{j=1}^n (2\pi g_j) \delta(y - x_j). \tag{3.6}$$

Both for feature regime and NTK limit, the first derivative of $\tilde{\varphi}(x)$ is continuous except for two values of x (0 and π for lazy, $-\pi/2$ and $\pi/2$ for feature), so that $\tilde{\varphi}(x)''$ has a singular part consisting of two Dirac delta functions.

As a result, the second derivative of the predictor $(f^n)''$ has a singular part consisting of many Dirac deltas. If we denote with $(f^n)''_r$ the regular part, obtained by subtracting all the delta functions, we can show that (see App. C):

Proposition 1. ***(informal)*** *As $n \to \infty$, $(f^n)''_r$ converges to a function having finite second moment, i.e.*

$$\lim_{n\to\infty} \mathbb{E}_{f^*}[(f^n)''_r(x)]^2 = \textit{const.} < \infty. \tag{3.7}$$

In the large n limit, the predictor displays a singular second derivative at $O(n)$ points. Proposition 1 implies that outside of these singular points the second derivative is well defined. Thus, as n gets large and the singular points approach each other, the predictor can be approximated by a chain of parabolas, as highlighted in Fig. 3 and noticed in [47] for a Laplace kernel. This property alone allows to determine the asymptotic scaling of the error in $d=2$. In simple terms, Prop. 1 follows from the convergence of g^n to the function satisfying $f^*(x) = \int \frac{dy}{2\pi} g(y)\tilde{\varphi}_r(x-y)$, which is guaranteed under our assumptions on the target function—a detailed proof is given in App. C.

Decay of the error in $d=2$ (sketch) The full calculation is in App. D. Consider a slightly simplified problem where $\tilde{\varphi}$ has a single discontinuity in its derivative, located at $x=0$. In this case, $f^n(x)$ is singular if and only if x is a data point. Consider then the interval $x \in [x_i, x_{i+1}]$ and set $\delta_i = x_{i+1} - x_i$, $x_{i+1/2} = (x_{i+1} + x_i)/2$. If the target function is smooth enough ($\nu_t > 2$), then a Taylor expansion implies $|f^*(x_{i+1/2}) - f^n(x_{i+1/2})| \sim \delta_i^2$. Since the distances δ_i between adjacent singular points are random variables with mean of order $1/n$ and finite moments, it is straightforward to obtain that $\bar{\epsilon}(n) \sim \sum_i (f^*(x_{i+1/2}) - f^n(x_{i+1/2}))^2 \sim \sum_i \delta_i^4 \sim n^{-4}$. By contrast if f^* is not sufficiently smooth ($\nu_t \leq 2$), then $|f^*(x_{i+1/2}) - f^n(x_{i+1/2})| \sim \delta_i^{2\nu_t}$, leading to $\bar{\epsilon}(n) \sim n^{-2\nu_t}$. Note that for this asymptotic argument to apply to the feature learning regime, one must ensure that the distribution of the rescaled distance between adjacent singularities $n\delta_i$ has a finite fourth moment. This is obvious in the lazy regime, where the δ_i's are controlled by the position of the training points, but not in the feature regime, where the distribution of singular points is determined by that of the neuron's features. Nevertheless, we show that it must be the case in our setup in App. D.

Interpretation in terms of spectral bias From the discussion above it is evident that there is a length scale δ of order $1/n$ such that $f^n(x)$ is a good approximation of $f^*(x)$ over scales larger than δ. In terms of Fourier modes[2], one has: *i)* $\widehat{f^n}(k)$ matches $\widehat{f^n}(k)$ at long wavelengths, i.e. for $k \ll k_c \sim 1/n$. *ii)* In addition, since the phases $\exp(ikx_j)$ become effectively random phases for $k \gg k_c$, $\widehat{g^n}(k) = \sum_j g_j \exp(ikx_j)$ becomes a Gaussian random variable with zero mean and fixed variance and thus *iii)* $\widehat{f^n}(k) = \widehat{g^n}(k)\widehat{\tilde{\varphi}}(k)$ decorrelates from f^* for $k \gg k_c$. Therefore

$$\bar{\epsilon}(n) \sim \sum_{|k|>k_c} \mathbb{E}_{f^*}\left[\left(\widehat{g^n}(k)\widehat{\tilde{\varphi}}(k) - \widehat{f^n}(k)\right)^2\right] \sim \sum_{|k|\geq k_c} \mathbb{E}_{f^*}\left[(\widehat{g^n}(k))^2\right]\widehat{\tilde{\varphi}}(k)^2 + \mathbb{E}_{f^*}\left[(\widehat{f^n}(k))^2\right]. \tag{3.8}$$

For $\nu_t > 2$, one has $\sum_j g_j^2 \sim n^{-1} \lim_{n\to\infty} \int g^n(x)^2 dx \sim n^{-1}$. It follows (see App. E for details) that the sum is dominated by the first term, hence entirely controlled by the Fourier coefficients of $\widehat{f^n}(k)$ at large k. A smoother predictor corresponds to a faster decay of $\widehat{f^n}(k)$ with k, thus a faster decay of the error with n. Plugging the relevant decays yields $\bar{\epsilon} \sim n^{-4}$ for feature regime and lazy regime with the NTK, and n^{-6} for lazy regime with the RFK (which is smoother than the NTK). For $\nu_t \leq 2$, the two terms have comparable magnitude (see App. E), thus $\bar{\epsilon} \sim n^{-2\nu_t}$.

Generalization to higher dimensions The argument above can be generalized for any d by replacing Fourier modes with projections onto spherical harmonics. The characteristic distance between training points scales as $n^{-1/(d-1)}$, thus $k_c \sim n^{-1/(d-1)}$. Our ansatz is that, as in $d=2$: *i)* for $k \ll k_c$, the predictor modes coincide with those of the target function, $f^n_{k,l} \approx f^*_{k,l}$ (this corresponds to the spectral bias result of kernel methods, stating that the predictor reproduces the first $O(n)$ projections of the target in the kernel eigenbasis [45]); *ii)* For $k \gg k_c$, $g^n_{k,l}$ is a sum of uncorrelated terms, thus a Gaussian variable with zero mean and fixed variance; *iii)* $f^n_{k,\ell} = g^n_{k,\ell}\tilde{\varphi}_k$ decorrelates from $f^*_{k,\ell}$ for $k \gg k_c$. *i)*, *ii)* and *iii)* imply that:

$$\bar{\epsilon}(n) \sim \sum_{k\geq k_c} \sum_{l=1}^{\mathcal{N}_{k,d}} \mathbb{E}_{f^*}\left[\left(f^n_{k,l} - f^*_{k,l}\right)^2\right] \sim \sum_{k\geq k_c} \sum_{l=1}^{\mathcal{N}_{k,d}} \mathbb{E}_{f^*}\left[(g^n_{k,l})^2\right]\varphi_k^2 + k^{-2\nu_t-(d-1)}. \tag{3.9}$$

As shown in App. E, from this expression it is straightforward to obtain Eq. 3.4. Notice again that when the target is sufficiently smooth so that the predictor-dependent term dominates, the error is determined by the smoothness of the predictor. In particular, as $d > 2$, the predictor of feature learning is less smooth than both the NTK and RFK ones, due to the slower decay of the corresponding φ_k.

[2] The Fourier transform of a function $f(x)$ is indicated by the hat, $\widehat{f}(k)$.

4 Numerical tests of the theory

We test successfully our predictions by computing the learning curves of both lazy and feature regimes when *(i)* the target function is constant on the sphere for varying d, see Fig. 4, and *(ii)* the target is a Gaussian random field with varying smoothness ν_t, as shown in Fig. G.1 of App. G. For the lazy regime, we perform kernel regression using the analytical expression of the NTK [52] (see also Eq. A.19). For the feature regime, we find that our predictions hold when having a small regularization, although it takes unreachable times for gradient descent to exactly recover the minimal-norm solution—a more in-depth discussion can be found in App. G. An example of the atomic distribution of neurons found after training, which contrasts with the initial distribution, is displayed in Fig. 5a, left panel.

Another way to obtain sparse features is to initialize the network with very small weights [14], as proposed in [8]. As in the presence of an infinitesimal weights decay, this scheme also leads to sparse solutions with $n_A = \mathcal{O}(n)$ – an asymptotic dependence confirmed in Fig. G.3 of App. G. This observation implies that our predictions must apply in that case too, as we confirm in Fig. G.3.

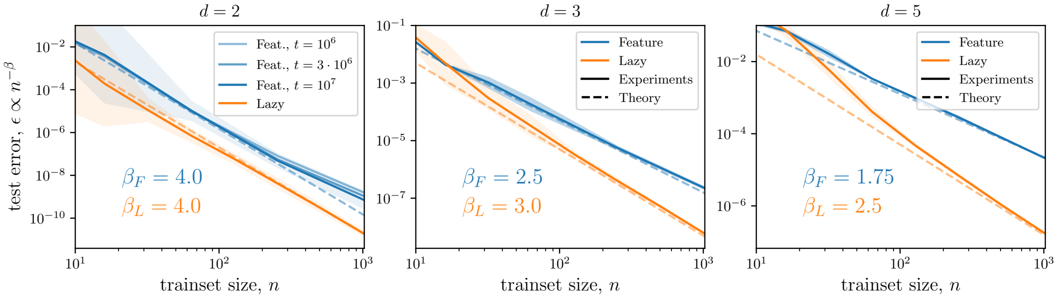


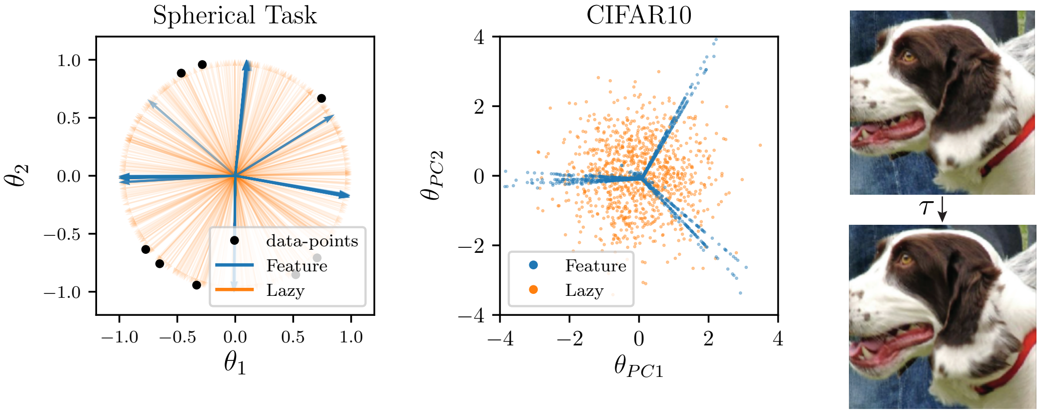
Figure 4: **Generalization error for a constant function $f^*(\boldsymbol{x}) = 1$.** Generalization error as a function of the training set size n for a network trained in the feature regime with L1 regularization (blue) and kernel regression corresponding to the infinite-width lazy regime (orange). Numerical results (full lines) and the exponents predicted by the theory (dashed) are plotted. Panels correspond to different input-space dimensions ($d = 2, 3, 5$). Results are averaged over 10 different initializations of the networks and datasets. For $d = 2$ and large n, the gap between experiments and prediction for the feature regime is due to the finite training time t. Indeed our predictions become more accurate as t increases, as illustrated in the left.

5 Evidence for overfitting along diffeomorphisms in image datasets

For fully-connected networks, the feature regime is well-adapted to learn anisotropic tasks [16]: if the target function does not depend on a certain linear subspace of input space, e.g. the pixels at the corner of an image, then neurons align perpendicularly to these directions [19]. By contrast, our results highlight a drawback of this regime when the target function is constant or smooth along directions in input space that require a continuous distribution of neurons to be represented. In such a case, the adaptation of the weights to the training points leads to a predictor with a sparse representation. Such a predictor would be less smooth than in the lazy regime and thus underperform.

Does this view hold for images, and explain why learning their features is detrimental for fully-connected networks? The first positive empirical evidence is that the neurons' distribution of networks trained on image data becomes indeed sparse in the feature regime, as illustrated in Fig. 5a, right, for CIFAR10 [28]. This observation raises the question of which are the directions in input space *i)* along which the target should vary smoothly, and *ii)* that are not easily represented by a discrete set of neurons. An example of such directions are global translations, which conserve the norm of the input and do not change the image class: the lazy regime predictor is indeed smoother than the feature one with respect to translations of the input (see App. H). Yet, these transformations live in a space of dimension 2, which is small in comparison with the full dimensionality d of the data and thus may play a negligible role.

A much larger class of transformations believed to have little effect on the target are small diffeomorphisms [29]. A diffeomorphism τ acting on an image is illustrated in Fig. 5b, which highlights that



(a) **Features sparsification.** 1$^{\text{st}}$Panel: Distribution of neuron's feature for the task of learning a constant function on the sphere in 2D. Arrows represent a subset of the network features $\{\boldsymbol{\theta}_h\}_{h=1}^{H}$ after training in the lazy and feature regimes. Training is performed on $n = 8$ data-points (black dots). 2$^{\text{nd}}$Panel: FCN trained on CIFAR10. On the axes the first two principal components of the features $\{\boldsymbol{\theta}_h\}_{h=1}^{H}$ after training on $n = 32$ points in the feature (blue) and lazy (orange) regimes. Similarly to what is observed when learning a constant function, the $\boldsymbol{\theta}_h$ angular distribution becomes sparse with training in the feature regime.

(b) **Example of diffeomorphism.** Sample of a max-entropy deformation τ [49] when applied to a natural image, illustrating that it does not change the image class for the human brain.

Figure 5: **Features sparsification and example of a diffeomorphism.**

our brain still perceives the content of the transformed image as in the original one. Near-invariance of the task to these transformations is believed to play a key role in the success of deep learning, and in explaining how neural networks beat the curse of dimensionality [48]. Indeed, if modern architectures can become insensitive to these transformations, then the dimensionality of the problem is considerably reduced. In fact, it was found that the architectures displaying the best performance are precisely those which learn to vary smoothly along such transformations [49].

Small diffeomorphisms are likely the directions we are looking for. To test this hypothesis, following [49], we characterize the smoothness of a function along such diffeomorphisms, relative to that of random directions in input space. Specifically, we use the *relative sensitivity*:

$$R_f = \frac{\mathbb{E}_{x,\tau}\|f(\tau x) - f(x)\|^2}{\mathbb{E}_{x,\eta}\|f(x+\eta) - f(x)\|^2}. \tag{5.1}$$

In the numerator, the average is made over the test set and over an ensemble of diffeomorphisms, reviewed in App. I. The magnitude of the diffeomorphisms is chosen so that each pixel is shifted by one on average. In the denominator, the average runs over the test set and the vectors η sampled uniformly on the sphere of radius $\|\eta\| = \mathbb{E}_{x,\tau}\|\tau x - x\|$, and this fixes the transformations magnitude.

We measure R_f as a function of n for three benchmark datasets of images, as shown in Fig. 6. We indeed find that R_f is consistently smaller in the lazy training regime, where features are not learned. Overall, this observation supports the view that learning sparse features is detrimental when data present (near) invariance to transformations that cannot be represented sparsely by the architecture considered. Fig. 1 supports the idea that—for benchmark image datasets—this negative effect overcomes well-known positive effects of learning features, e.g. becoming insensitive to pixels on the edge of images (see App. H for evidence of this effect).

6 Conclusion

Our central result is that learning sparse features can be detrimental if the task presents invariance or smooth variations along transformations that are not adequately captured by the neural network architecture. For fully-connected networks, these transformations can be rotations of the input, but also continuous translations and diffeomorphisms.

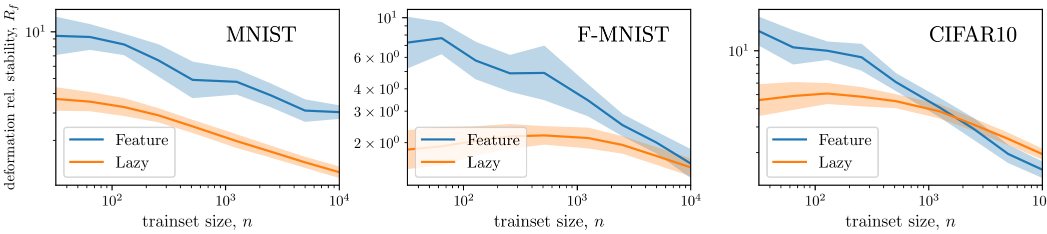


Figure 6: **Sensitivity to diffeomorphisms vs number of training points.** Relative sensitivity of the predictor to small diffeomorphisms of the input images, in the two regimes, for varying number of training points n and different image datasets. Smaller values correspond to a smoother predictor, on average. Results are computed using the same predictors as in Fig. 1.

Our analysis relies on the sparsity of the features learned by a shallow fully-connected architecture: even in the infinite width limit, when trained in the feature learning regime such networks behave as $\mathcal{O}(n)$ neurons. The asymptotic analysis we perform for random Gaussian fields on the sphere leads to predictions for the learning curve exponent β in different training regimes, which we verify. Such kind of results is scarce in the literature.

Note that our analysis focuses on ReLU neurons because *(i)* these are very often used in practice and *(ii)* in that case, β will depend on the training regime, allowing for stringent numerical tests. If smooth activations (e.g. softplus) are considered, we expect that learning features will still be detrimental for generalization. Yet, the difference will not appear in the exponent β, but in other aspects of the learning curves (including numerical coefficients and pre-asymptotic effects) that are harder to predict.

Most fundamentally, our results underline that the success of feature learning for modern architectures still lacks a sufficient explanation. Indeed, most of the theoretical studies that previously emphasized the benefits of learning features have been considering fully-connected networks, for which learning features can be in practice a drawback. It is tempting to argue that in modern architectures, learning features is not at a disadvantage because smoothness along diffeomorphisms can be enforced from the start—due to the locally connected, convolutional, and pooling layers [53, 29]. Yet the best architectures often do not perform pooling and are not stable toward diffeomorphisms at initialization. *During training*, learning features leads to more stable and smoother solutions along diffeomorphisms [54, 49]. Understanding why building sparse features enhances stability in these architectures may ultimately explain the magical feat of deep CNNs: learning tasks in high dimensions.

Acknowledgements

We thank Lénaïc Chizat, Antonio Sclocchi, and Umberto M. Tomasini for helpful discussions. The work of MW is supported by a grant from the Simons Foundation (#454953) and from the NSF under Grant No. 200021-165509. The work of EVE is supported by the National Science Foundation under awards DMR-1420073, DMS-2012510, and DMS-2134216, by the Simons Collaboration on Wave Turbulence, Grant No. 617006, and by a Vannevar Bush Faculty Fellowship.

References

[1] Ulrike von Luxburg and Olivier Bousquet. Distance-based classification with lipschitz functions. *Journal of Machine Learning Research*, 5(Jun):669–695, 2004.

[2] Francis Bach. Breaking the curse of dimensionality with convex neural networks. *The Journal of Machine Learning Research*, 18(1):629–681, 2017.

[3] Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md Patwary, Mostofa Ali, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, empirically. *arXiv preprint arXiv:1712.00409*, 2017.

[4] Quoc V Le. Building high-level features using large scale unsupervised learning. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 8595–8598. IEEE, 2013.

[5] Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

[6] Alessio Ansuini, Alessandro Laio, Jakob H Macke, and Davide Zoccolan. Intrinsic dimension of data representations in deep neural networks. In *Advances in Neural Information Processing Systems*, pages 6111–6122, 2019.

[7] Stefano Recanatesi, Matthew Farrell, Madhu Advani, Timothy Moore, Guillaume Lajoie, and Eric Shea-Brown. Dimensionality compression and expansion in deep neural networks. *arXiv preprint arXiv:1906.00443*, 2019.

[8] Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. In *Advances in Neural Information Processing Systems*, pages 2937–2947, 2019.

[9] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Proceedings of the 32Nd International Conference on Neural Information Processing Systems*, NIPS'18, pages 8580–8589, USA, 2018. Curran Associates Inc.

[10] Simon S. Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. In *International Conference on Learning Representations*, 2019.

[11] Grant M Rotskoff and Eric Vanden-Eijnden. Neural networks as interacting particle systems: Asymptotic convexity of the loss landscape and universal scaling of the approximation error. *arXiv preprint arXiv:1805.00915*, 2018.

[12] Song Mei, Andrea Montanari, and Phan-Minh Nguyen. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33):E7665–E7671, August 2018. Publisher: National Academy of Sciences Section: PNAS Plus.

[13] Justin Sirignano and Konstantinos Spiliopoulos. Mean Field Analysis of Neural Networks: A Law of Large Numbers. *SIAM Journal on Applied Mathematics*, 80(2):725–752, January 2020. Publisher: Society for Industrial and Applied Mathematics.

[14] Blake Woodworth, Suriya Gunasekar, Jason D. Lee, Edward Moroshko, Pedro Savarese, Itay Golan, Daniel Soudry, and Nathan Srebro. Kernel and Rich Regimes in Overparametrized Models. In *Conference on Learning Theory*, pages 3635–3673. PMLR, July 2020. ISSN: 2640-3498.

[15] Jaume de Dios and Joan Bruna. On sparsity in overparametrised shallow relu networks. *arXiv preprint arXiv:2006.10225*, 2020.

[16] Lénaïc Chizat and Francis Bach. Implicit Bias of Gradient Descent for Wide Two-layer Neural Networks Trained with the Logistic Loss. In *Conference on Learning Theory*, pages 1305–1338. PMLR, July 2020. ISSN: 2640-3498.

[17] Behrooz Ghorbani, Song Mei, Theodor Misiakiewicz, and Andrea Montanari. When Do Neural Networks Outperform Kernel Methods? *Advances in Neural Information Processing Systems*, 33, 2020.

[18] Maria Refinetti, Sebastian Goldt, Florent Krzakala, and Lenka Zdeborová. Classifying high-dimensional gaussian mixtures: Where kernel methods fail and neural networks succeed. *arXiv preprint arXiv:2102.11742*, 2021.

[19] Jonas Paccolat, Leonardo Petrini, Mario Geiger, Kevin Tyloo, and Matthieu Wyart. Geometric compression of invariant manifolds in neural networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(4):044001, April 2021. Publisher: IOP Publishing.

[20] Mario Geiger, Stefano Spigler, Arthur Jacot, and Matthieu Wyart. Disentangling feature and lazy training in deep neural networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(11):113301, November 2020. Publisher: IOP Publishing.

[21] Jaehoon Lee, Samuel S Schoenholz, Jeffrey Pennington, Ben Adlam, Lechao Xiao, Roman Novak, and Jascha Sohl-Dickstein. Finite versus infinite neural networks: an empirical study. *arXiv preprint arXiv:2007.15801*, 2020.

[22] Guillermo Ortiz-Jiménez, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. What can linearized neural networks actually say about generalization? *Advances in Neural Information Processing Systems*, 34, 2021.

[23] Yilan Chen, Wei Huang, Lam M. Nguyen, and Tsui-Wei Weng. On the Equivalence between Neural Network and Support Vector Machine. *arXiv:2111.06063 [cs, math, stat]*, November 2021. arXiv: 2111.06063.

[24] Mario Geiger, Arthur Jacot, Stefano Spigler, Franck Gabriel, Levent Sagun, Stéphane d'Ascoli, Giulio Biroli, Clément Hongler, and Matthieu Wyart. Scaling description of generalization with number of parameters in deep learning. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(2):023401, February 2020. Publisher: IOP Publishing.

[25] Mario Geiger, Leonardo Petrini, and Matthieu Wyart. Landscape and training regimes in deep learning. *Physics Reports*, 924:1–18, August 2021.

[26] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[27] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. *arXiv:1708.07747 [cs, stat]*, September 2017. arXiv: 1708.07747.

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

[29] Joan Bruna and Stéphane Mallat. Invariant scattering convolution networks. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1872–1886, 2013.

[30] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[31] Hartmut Maennel, Olivier Bousquet, and Sylvain Gelly. Gradient descent quantizes relu network features. *arXiv preprint arXiv:1803.08367*, 2018.

[32] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In *Conference on Learning Theory*, pages 1376–1401. PMLR, 2015.

[33] Claire Boyer, Antonin Chambolle, Yohann De Castro, Vincent Duval, Frédéric De Gournay, and Pierre Weiss. On Representer Theorems and Convex Regularization. *SIAM Journal on Optimization*, 29(2):1260–1281, May 2019.

[34] Lenaic Chizat. Sparse optimization on measures with over-parameterized gradient descent. *Mathematical Programming*, pages 1–46, 2021.

[35] Bruno A. Olshausen and David J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, June 1996. Number: 6583 Publisher: Nature Publishing Group.

[36] Julien Mairal, Jean Ponce, Guillermo Sapiro, Andrew Zisserman, and Francis Bach. Supervised Dictionary Learning. In *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc., 2008.

[37] Nishant A Mehta and Alexander G Gray. Sparsity-Based Generalization Bounds for Predictive Sparse Coding. page 9.

[38] Jeremias Sulam, Ramchandran Muthukumar, and Raman Arora. Adversarial Robustness of Supervised Sparse Coding, January 2021. arXiv:2010.12088 [cs, stat].

[39] Gilad Yehudai and Ohad Shamir. On the power and limitations of random features for understanding neural networks. In *Advances in Neural Information Processing Systems*, pages 6598–6608, 2019.

[40] Behrooz Ghorbani, Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Limitations of lazy training of two-layers neural network. In *Advances in Neural Information Processing Systems*, pages 9111–9121, 2019.

[41] Vardan Papyan, XY Han, and David L Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663, 2020.

[42] Behnam Neyshabur. Towards Learning Convolutions from Scratch. *arXiv:2007.13657 [cs, stat]*, July 2020. arXiv: 2007.13657.

[43] Alessandro Ingrosso and Sebastian Goldt. Data-driven emergence of convolutional structure in neural networks. *Proceedings of the National Academy of Sciences*, 119(40):e2201854119, 2022.

[44] Stefano Spigler, Mario Geiger, and Matthieu Wyart. Asymptotic learning curves of kernel methods: empirical data versus teacher–student paradigm. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(12):124001, December 2020. Publisher: IOP Publishing.

[45] Blake Bordelon, Abdulkadir Canatar, and Cengiz Pehlevan. Spectrum Dependent Learning Curves in Kernel Regression and Wide Neural Networks. In *International Conference on Machine Learning*, pages 1024–1034. PMLR, November 2020. ISSN: 2640-3498.

[46] Hugo Cui, Bruno Loureiro, Florent Krzakala, and Lenka Zdeborová. Generalization error rates in kernel regression: The crossover from the noiseless to noisy regime. *Advances in Neural Information Processing Systems*, 34, 2021.

[47] Umberto M Tomasini, Antonio Sclocchi, and Matthieu Wyart. Failure and success of the spectral bias prediction for kernel ridge regression: the case of low-dimensional data. *arXiv preprint arXiv:2202.03348*, 2022.

[48] Stéphane Mallat. Understanding deep convolutional networks. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150203, 2016.

[49] Leonardo Petrini, Alessandro Favero, Mario Geiger, and Matthieu Wyart. Relative stability toward diffeomorphisms indicates performance in deep nets. In *Advances in Neural Information Processing Systems*, volume 34, pages 8727–8739. Curran Associates, Inc., 2021.

[50] Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1998.

[51] Bernhard Scholkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond.* MIT press, 2001.

[52] Youngmin Cho and Lawrence K. Saul. Kernel methods for deep learning. In *Advances in Neural Information Processing Systems 22*, pages 342–350. Curran Associates, Inc., 2009.

[53] Alberto Bietti and Julien Mairal. Group invariance, stability to deformations, and complexity of deep convolutional representations. *The Journal of Machine Learning Research*, 20(1):876–924, 2019.

[54] Avraham Ruderman, Neil C. Rabinowitz, Ari S. Morcos, and Daniel Zoran. Pooling is neither necessary nor sufficient for appropriate deformation stability in CNNs. *arXiv:1804.04438 [cs, stat]*, May 2018. arXiv: 1804.04438.

[55] Alex Smola, Zoltán Ovári, and Robert C. Williamson. Regularization with dot-product kernels. *Advances in neural information processing systems*, 13, 2000.

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

[57] Costas Efthimiou and Christopher Frye. *Spherical harmonics in p dimensions*. World Scientific, 2014.

[58] Alberto Bietti and Francis Bach. Deep Equals Shallow for ReLU Networks in Kernel Regimes. *arXiv:2009.14397 [cs, stat]*, August 2021. arXiv: 2009.14397.

[59] Francis Bach. Learning Theory from First Principles. *In preparation*, 2022.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] Yes, in the conclusion we state that this paper provides a better understanding of the feature regime for FCNs while a better understanding of its advantages in other architectures like CNNs is still lacking.
   (c) Did you discuss any potential negative societal impacts of your work? [No] Our work is theoretical in nature and we don't foresee possible negative societal consequences of our work.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] Yes, proofs are either provided in the main text or in appendix, depending on space constraints.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We provide links to the (anonymized) github repository, code and instructions on how to run it are found there.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Training details are either in the text or in the experiments.md file, as indicated in App. G.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Statistical errors are reported as shaded area around the curves.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Details are provided in App. G.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We use the PyTorch framework and cite the authors in the text.
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] The code written by the authors for running experiments is available as a URL.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Quick recap of spherical harmonics

Spherical harmonics This appendix collects some introductory background on spherical harmonics and dot-product kernels on the sphere [55]. See [56, 57] for an expanded treatment. Spherical harmonics are homogeneous polynomials on the sphere $\mathbb{S}^{d-1}=\{\boldsymbol{x} \in \mathbb{R}^d \,|\, \|\boldsymbol{x}\|=1\}$, with $\|.\|$ denoting the L2 norm. Given the polynomial degree $k \in \mathbb{N}$, there are $\mathcal{N}_{k,s}$ linearly independent spherical harmonics of degree k on $\mathbb{S}^{s-1}$, with

$$\mathcal{N}_{k,d} = \frac{2k+d-2}{k}\binom{d+k-3}{k-1}, \quad \begin{cases} \mathcal{N}_{0,d} = 1 \quad \forall d, \\ \mathcal{N}_{k,d} \asymp A_d k^{d-2} \quad \text{for } k \gg 1, \end{cases} \tag{A.1}$$

where $\asymp$ means logarithmic equivalence for $k \to \infty$ and $A_d = \sqrt{2/\pi}(d-2)^{\frac{3}{2}-d}e^{d-2}$. Thus, we can introduce a set of $\mathcal{N}_{k,d}$ spherical harmonics $Y_{k,\ell}$ for each k, with ℓ ranging in $1,\dots,\mathcal{N}_{k,d}$, which are orthonormal with respect to the uniform measure on the sphere $d\tau(\boldsymbol{x})$,

$$\{Y_{k,\ell}\}_{k\geq 0, \ell=1,\dots,\mathcal{N}_{k,d}}\,, \quad \langle Y_{k,\ell}, Y_{k,\ell'}\rangle_{\mathbb{S}^{d-1}} := \int_{\mathbb{S}^{d-1}} Y_{k,\ell}(\boldsymbol{x})Y_{k,\ell'}(\boldsymbol{x})\, d\tau(\boldsymbol{x}) = \delta_{\ell,\ell'}. \tag{A.2}$$

Because of the orthogonality of homogeneous polynomials with different degree, the set is a complete orthonormal basis for the space of square-integrable functions on $\mathbb{S}^{d-1}$. For any function $f: \mathbb{S}^{d-1} \to \mathbb{R}$, then

$$f(\boldsymbol{x}) = \sum_{k\geq 0}\sum_{\ell=1}^{\mathcal{N}_{k,d}} f_{k,\ell}Y_{k,\ell}(\boldsymbol{x}), \quad f_{k,\ell} = \int_{\mathbb{S}^{d-1}} f(\boldsymbol{x})Y_{k,\ell}(\boldsymbol{x})d\tau(\boldsymbol{x}). \tag{A.3}$$

Furthermore, spherical harmonics are eigenfunctions of the Laplace-Beltrami operator Δ, which is nothing but the restriction of the standard Laplace operator to $\mathbb{S}^{d-1}$,

$$\Delta Y_{k,\ell} = -k(k+d-2)Y_{k,\ell}. \tag{A.4}$$

Legendre polynomials By fixing a direction $\boldsymbol{y}$ in $\mathbb{S}^{d-1}$ one can select, for each k, the only spherical harmonic of degree k which is invariant for rotations that leave $\boldsymbol{y}$ unchanged. This particular spherical harmonic is, in fact, a function of $\boldsymbol{x}\cdot\boldsymbol{y}$ and is called the Legendre polynomial of degree k, $P_{k,d}(\boldsymbol{x}\cdot\boldsymbol{y})$ (also referred to as Gegenbauer polynomial). Legendre polynomials can be written as a combination of the orthonormal spherical harmonics $Y_{k,\ell}$ via the addition theorem [56, Thm. 2.9],

$$P_{k,d}(\boldsymbol{x}\cdot\boldsymbol{y}) = \frac{1}{\mathcal{N}_{k,d}}\sum_{\ell=1}^{\mathcal{N}_{k,d}} Y_{k,\ell}(\boldsymbol{x})Y_{k,\ell}(\boldsymbol{y}). \tag{A.5}$$

Alternatively, $P_{k,d}$ is given explicitly as a function of $t=\boldsymbol{x}\cdot\boldsymbol{y} \in [-1,1]$ via the Rodrigues' formula [56, Thm. 2.23],

$$P_{k,d}(t) = \left(-\frac{1}{2}\right)^k \frac{\Gamma\left(\frac{d-1}{2}\right)}{\Gamma\left(k+\frac{d-1}{2}\right)}\left(1-t^2\right)^{\frac{3-d}{2}}\frac{d^k}{dt^k}\left(1-t^2\right)^{k+\frac{d-3}{2}}. \tag{A.6}$$

Here Γ denotes the Gamma function, $\Gamma(z)=\int_0^\infty x^{z-1}e^{-x}\,dx$. Legendre polynomials are orthogonal on $[-1,1]$ with respect to the measure with density $(1-t^2)^{(d-3)/2}$, which is the probability density function of the scalar product between to points on $\mathbb{S}^{d-1}$.

$$\int_{-1}^{+1} P_{k,d}(t)P_{k',d}(t)\,\left(1-t^2\right)^{\frac{d-3}{2}}dt = \frac{|\mathbb{S}^{d-1}|}{|\mathbb{S}^{d-2}|}\frac{\delta_{k,k'}}{\mathcal{N}_{k,s}}. \tag{A.7}$$

Here $|\mathbb{S}^{d-1}|=2\pi^{\frac{d}{2}}/\Gamma(\frac{d}{2})$ denotes the surface area of the d-dimensional unit sphere ($|\mathbb{S}^0|=2$ by definition).

To sum up, given $\boldsymbol{x},\boldsymbol{y} \in \mathbb{S}^{d-1}$, functions of $\boldsymbol{x}$ or $\boldsymbol{y}$ can be expressed as a sum of projections on the orthonormal spherical harmonics, whereas functions of $\boldsymbol{x}\cdot\boldsymbol{y}$ can be expressed as a sum of projections on the Legendre polynomials. The relationship between the two expansions is elucidated in the Funk-Hecke formula [56, Thm. 2.22],

$$\int_{\mathbb{S}^{d-1}} f(\boldsymbol{x}\cdot\boldsymbol{y})Y_{k,\ell}(\boldsymbol{y})\,d\tau(\boldsymbol{y}) = Y_{k,\ell}(\boldsymbol{x})\frac{|\mathbb{S}^{d-2}|}{|\mathbb{S}^{d-1}|}\int_{-1}^{+1} f(t)P_{k,d}(t)\,\left(1-t^2\right)^{\frac{d-3}{2}}dt := f_k Y_{k,\ell}(\boldsymbol{x}). \tag{A.8}$$

A.1 Expansion of ReLU and combinations thereof

We can apply Eq. A.8 to have an expansion of neurons $\sigma\left(\boldsymbol{\theta}\cdot\boldsymbol{x}\right)$ in terms of spherical harmonics [2, Appendix D]. After defining

$$\varphi_k := \frac{|\mathbb{S}^{d-2}|}{|\mathbb{S}^{d-1}|}\int_{-1}^{+1}\sigma(t)P_{k,d}(t)\,\left(1-t^2\right)^{\frac{d-3}{2}}dt, \tag{A.9}$$

one has

$$\sigma\left(\boldsymbol{\theta}\cdot\boldsymbol{x}\right) = \sum_{k\geq 0}\mathcal{N}_{k,d}\varphi_k P_{k,d}\left(\boldsymbol{\theta}\cdot\boldsymbol{x}\right) = \sum_{k\geq 0}\varphi_k\sum_{\ell=1}^{\mathcal{N}_{k,d}}Y_{k,\ell}(\boldsymbol{\theta})Y_{k,\ell}(\boldsymbol{x}). \tag{A.10}$$

For ReLU activations, in particular, $\sigma(t) = \max(0,t)$, thus

$$\varphi_k^{\mathrm{ReLU}} = \frac{|\mathbb{S}^{d-2}|}{|\mathbb{S}^{d-1}|}\int_{0}^{+1}tP_{k,d}(t)\,\left(1-t^2\right)^{\frac{d-3}{2}}dt. \tag{A.11}$$

Notice that when k is odd $P_{k,d}$ is an odd function of t, thus the integrand $tP_{k,d}(t)(1-t^2)^{\frac{d-3}{2}}$ is an even function of t. As a result the integral on the right-hand side of Eq. A.11 coincides with half the integral over the full domain $[-1,1]$,

$$\int_{0}^{+1}tP_{k,d}(t)\,\left(1-t^2\right)^{\frac{d-3}{2}}dt = \frac{1}{2}\int_{-1}^{+1}tP_{k,d}(t)\,\left(1-t^2\right)^{\frac{d-3}{2}}dt = 0 \text{ for } k>1, \tag{A.12}$$

because, due to Eq. A.7, $P_{k,d}$ is orthogonal to all polynomials with degree strictly lower than k. For even k we can use Eq. A.6 and get [2] (see Eq. 3.3, main text)

$$\begin{aligned}\int_{0}^{+1}tP_{k,d}(t)\,\left(1-t^2\right)^{\frac{d-3}{2}}dt &= \left(-\frac{1}{2}\right)^k\frac{\Gamma\left(\frac{d-1}{2}\right)}{\Gamma\left(k+\frac{d-1}{2}\right)}\int_0^1 t\frac{d^k}{dt^k}\left(1-t^2\right)^{k+\frac{d-3}{2}}\,dt\\ &= -\left(-\frac{1}{2}\right)^k\frac{\Gamma\left(\frac{d-1}{2}\right)}{\Gamma\left(k+\frac{d-1}{2}\right)}\frac{d^{k-2}}{dt^{k-2}}\left(1-t^2\right)^{k+\frac{d-3}{2}}\Bigg|_{t=0}^{t=1}\\ &\Rightarrow \varphi_k^{\mathrm{ReLU}} \sim k^{-\frac{d-1}{2}-\frac{3}{2}} \text{ for } k\gg 1 \text{ and even.}\end{aligned} \tag{A.13}$$

Because all $\varphi_k^{\mathrm{ReLU}}$ with $k>1$ and odd vanish, even summing an infinite amount of neurons $\sigma(\boldsymbol{\theta}\cdot\boldsymbol{x})$ with varying $\boldsymbol{\theta}$ does not allow to approximate any function on $\mathbb{S}^{d-1}$, but only those which have vanishing projections on all the spherical harmonics $Y_{k,\ell}$ with $k>1$ and odd. This is why we set the odd coefficients of the target function spectrum to zero in Eq. 2.1.

A.2 Dot-product kernels on the sphere

Also general dot-product kernels on the sphere admit an expansion such as Eq. A.10,

$$\mathcal{C}\left(\boldsymbol{x}\cdot\boldsymbol{y}\right) = \sum_{k\geq 0}\mathcal{N}_{k,d}c_k P_{k,d}\left(\boldsymbol{\theta}\cdot\boldsymbol{x}\right) = \sum_{k\geq 0}c_k\sum_{\ell=1}^{\mathcal{N}_{k,d}}Y_{k,\ell}(\boldsymbol{\theta})Y_{k,\ell}(\boldsymbol{x}), \tag{A.14}$$

with

$$c_k = \frac{|\mathbb{S}^{d-2}|}{|\mathbb{S}^{d}|}\int_{-1}^{1}\mathcal{C}(t)P_{k,d}(t)\,\left(1-t^2\right)^{\frac{d-3}{2}}dt. \tag{A.15}$$

The asymptotic decay of c_k for large k is controlled by the behaviour of $\mathcal{C}(t)$ near $t=\pm 1$, [58]. More precisely [58, Thm. 1], if $\mathcal{C}$ is infinitely differentiable in $(-1,1)$ and has the following expansion around ± 1,

$$\begin{cases}\mathcal{C}(t) = p_1(1-t) + c_1(1-t)^\nu + o\left((1-t)^\nu\right) \text{ near } t=+1;\\ \mathcal{C}(t) = p_{-1}(-1+t) + c_{-1}(-1+t)^\nu + o\left((-1+t)^\nu\right) \text{ near } t=-1,\end{cases} \tag{A.16}$$

where $p_{\pm 1}$ are polynomials and ν is not an integer, then

$$\begin{aligned}k \text{ even: } c_k &\sim (c_1+c_{-1})k^{-2\nu-(d-1)};\\ k \text{ odd: } c_k &\sim (c_1-c_{-1})k^{-2\nu-(d-1)},\end{aligned} \tag{A.17}$$

The result above implies that that if $c_1 = c_{-1}$ ($c_1 = -c_{-1}$), then the eigenvalues with k odd (even) decay faster than $k^{-2\nu-(d-2)}$. Moreover, if $\mathcal{C}$ is infinitely differentiable in $[-1,1]$ then c_k decays faster than any polynomial.

NTK and RFK of one-hidden-layer ReLU networks Let $\mathbb{E}_{\boldsymbol{\theta}}$ denote expectation over a multivariate normal distribution with zero mean and unitary covariance matrix. For any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{S}^{d-1}$, the RFK of a one-hidden-layer ReLU network Eq. 2.3 with all parameters initialised as independent Gaussian random numbers with zero mean and unit variance reads

$$\begin{aligned} K^{\text{RFK}}(\boldsymbol{x}\cdot\boldsymbol{y}) &= \mathbb{E}_{\boldsymbol{\theta}}\left[\sigma(\boldsymbol{\theta}\cdot\boldsymbol{x})\sigma(\boldsymbol{\theta}\cdot\boldsymbol{y})\right] \\ &= \frac{(\pi - \arccos(t))t + \sqrt{1-t^2}}{2\pi}, \text{ with } t = \boldsymbol{x}\cdot\boldsymbol{y}. \end{aligned} \tag{A.18}$$

The NTK of the same network reads, with σ' denoting the derivative of ReLU or Heaviside function,

$$\begin{aligned} K^{\text{NTK}}(\boldsymbol{x}\cdot\boldsymbol{y}) &= \mathbb{E}_{\boldsymbol{\theta}}\left[\sigma(\boldsymbol{\theta}\cdot\boldsymbol{x})\sigma(\boldsymbol{\theta}\cdot\boldsymbol{y})\right] + (\boldsymbol{x}\cdot\boldsymbol{y})\mathbb{E}_{\boldsymbol{\theta}}\left[\sigma'(\boldsymbol{\theta}\cdot\boldsymbol{x})\sigma'(\boldsymbol{\theta}\cdot\boldsymbol{y})\right] \\ &= \frac{2(\pi - \arccos(t))t + \sqrt{1-t^2}}{2\pi}, \text{ with } t = \boldsymbol{x}\cdot\boldsymbol{y}. \end{aligned} \tag{A.19}$$

As functions of a dot-product on the sphere, both NTK and RFK admit a decomposition in terms of spherical harmonics as Eq. A.15. For dot-product kernels, this expansion coincides with the Mercer's decomposition of the kernel [55], that is the coefficients of the expansion are the eigenvalues of the kernel. The asymptotic decay of the eigenvalues of such kernels φ_k^{NTK} and φ_k^{RFK} can be obtained by applying Eq. A.16 [58, Thm. 1]. Equivalently, one can notice that K^{RFK} is proportional to the convolution on the sphere of ReLU with itself, therefore $\varphi_k^{\text{RFK}} = (\varphi_k^{\text{ReLU}})^2$. Similarly, the asymptotic decay of φ_k^{NTK} can be related to that of the coefficients of σ', derivative of ReLU: $\varphi_k(\sigma') \sim k\varphi(\sigma)$, thus $\varphi_k^{\text{NTK}} \sim k^2(\varphi_k^{\text{ReLU}})^2$. Both methods lead to Eq. 3.3 of the main text.

Gaussian random fields and Eq. 2.2 Consider a Gaussian random field f^* on the sphere with covariance kernel $\mathcal{C}(\boldsymbol{x}\cdot\boldsymbol{y})$,

$$\mathbb{E}\left[f^*(\boldsymbol{x})\right] = 0, \quad \mathbb{E}\left[f^*(\boldsymbol{x})f^*(\boldsymbol{y})\right] = \mathcal{C}(\boldsymbol{x}\cdot\boldsymbol{y}), \quad \forall \boldsymbol{x},\boldsymbol{y} \in \mathbb{S}^{d-1}. \tag{A.20}$$

f^* can be equivalently specified via the statistics of the coefficients $f^*_{k,\ell}$,

$$\mathbb{E}\left[f^*_{k,\ell}\right] = 0, \quad \mathbb{E}\left[f^*_{k,\ell}f^*_{k',\ell'}\right] = c_k\delta_{k,k'}\delta_{\ell,\ell'}, \tag{A.21}$$

with c_k denoting the eigenvalues of $\mathcal{C}$ in Eq. A.15. Notice that the eigenvalues are degenerate with respect to ℓ because the covariance kernel is a function $\boldsymbol{x}\cdot\boldsymbol{y}$: as a result, the random function f^* is isotropic in law.

If c_k decays as a power of k, then such power controls the weak differentiability (in the mean-squared sense) of the random field f^*. In fact, from Eq. A.4,

$$\left\|\Delta^{m/2}f^*\right\| = \sum_{k\geq 0}\sum_{\ell}\left(-k(k+d-2)\right)^m\left(f^*_{k,\ell}\right)^2. \tag{A.22}$$

Upon averaging over f^* one gets

$$\mathbb{E}\left[\left\|\Delta^{m/2}f^*\right\|\right] = \sum_{k\geq 0}\left(-k(k+d-2)\right)^m\sum_{\ell}\mathbb{E}\left[\left(f^*_{k,\ell}\right)^2\right] = \sum_{k\geq 0}\left(-k(k+d-2)\right)^m\mathcal{N}_{k,d}c_k. \tag{A.23}$$

From Eq. A.16 [58, Thm. 1], if $\mathcal{C}(t) \sim (1-t)^{\nu_t}$ for $t \to 1$ and/or $\mathcal{C}(t) \sim (-1+t)^{\nu_t}$ for $t \to -1$, then $c_k \sim k^{-2\nu_t-(d-1)}$ for $k \gg 1$. In addition, for finite but arbitrary d, $\left(-k(k+d-2)\right)^m \sim k^{2m}$ and $\mathcal{N}_{k,s} \sim k^{d-2}$ (see Eq. A.1). Hence the summand in the right-hand side of Eq. A.23 is $\sim k^{2(m-\nu_t)-1}$, thus

$$\mathbb{E}\left[\left\|\Delta^{m/2}f^*\right\|\right] < \infty \quad \forall m < \nu_t. \tag{A.24}$$

Alternatively, one can think of ν_t as controlling the scaling of the difference δf^* over inputs separated by a distance δ. From Eq. A.20,

$$\begin{aligned} \mathbb{E}\left[|f^*(\boldsymbol{x}) - f^*(\boldsymbol{y})|^2\right] &= 2\mathcal{C}(1) - 2\mathcal{C}(\boldsymbol{x}\cdot\boldsymbol{y}) = 2\mathcal{C}(1) + O((1-\boldsymbol{x}\cdot\boldsymbol{y})^{\nu_t}) \\ &= 2\mathcal{C}(1) + O(|\boldsymbol{x}-\boldsymbol{y}|^{2\nu_t}) \end{aligned} \tag{A.25}$$

B Uniqueness and Sparsity of the L1 minimizer

Recall that we want to find the γ^* that solves

$$\gamma^* = \arg\min_{\gamma} \int_{\mathbb{S}^{d-1}} |d\gamma(\boldsymbol{\theta})| \quad \text{subject to} \quad \int_{\mathbb{S}^{d-1}} \sigma(\boldsymbol{\theta}\cdot\boldsymbol{x}_i)d\gamma(\boldsymbol{\theta}){=}f^*(\boldsymbol{x}_i) \quad \forall i=1,\dots,n. \tag{B.1}$$

In this appendix, we argue that the uniqueness of γ^* which implies that it is atomic with at most n atoms is a natural assumption. We start by discretizing the measure γ into H atoms, with H arbitrarily large. Then the problem Eq. B.1 can be rewritten as

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \|\boldsymbol{w}\|_1, \quad \text{subject to} \quad \boldsymbol{\Phi}\boldsymbol{w}=\boldsymbol{y}, \tag{B.2}$$

with $\boldsymbol{\Phi}\in\mathbb{R}^{H\times n}$, $\Phi_{h,i}=\sigma(\boldsymbol{\theta}_h\cdot\boldsymbol{x}_i)$ and $y_i=f^*(\boldsymbol{x}_i)$.

Given $\boldsymbol{w}\in\mathbb{R}^H$, let $\boldsymbol{u}=\max(\boldsymbol{w},0)\geq\boldsymbol{0}$ and $\boldsymbol{v}=-\max(-\boldsymbol{w},0)\geq\boldsymbol{0}$ so that $\boldsymbol{w}=\boldsymbol{u}-\boldsymbol{v}$. It is well-known (see e.g. [50]) that the minimization problem in (B.2) can be recast in terms of $\boldsymbol{u}$ and $\boldsymbol{v}$ into a linear programming problem. That is, $\boldsymbol{w}^*=\boldsymbol{u}^*-\boldsymbol{v}^*$ with

$$(\boldsymbol{u}^*,\boldsymbol{v}^*) = \arg\min_{\boldsymbol{u},\boldsymbol{v}} \boldsymbol{e}^T(\boldsymbol{u}+\boldsymbol{v}), \quad \text{subject to } \boldsymbol{\Phi}\boldsymbol{u}-\boldsymbol{\Phi}\boldsymbol{v}=\boldsymbol{y}, \quad \boldsymbol{u}\geq\boldsymbol{0}, \quad \boldsymbol{v}\geq\boldsymbol{0} \tag{B.3}$$

where $\boldsymbol{e}=[1,1,\dots,1]^T$. Assuming that this problem is feasible (i.e. there is at least one solution to $\boldsymbol{\Phi}\boldsymbol{u}-\boldsymbol{\Phi}\boldsymbol{v}=\boldsymbol{y}$ such that $\boldsymbol{u}\geq\boldsymbol{0}$, $\boldsymbol{v}\geq\boldsymbol{0}$), it is known that it admits extremal solution, i.e. solutions such that at most n entries of $(\boldsymbol{u}^*,\boldsymbol{v}^*)$ (and hence $\boldsymbol{w}^*$) are non-zero. The issue is whether such an extremal solution is unique. Assume that there are two, say $(\boldsymbol{u}_1^*,\boldsymbol{v}_1^*)$ and $(\boldsymbol{u}_2^*,\boldsymbol{v}_2^*)$. Then, by convexity,

$$(\boldsymbol{u}_t^*,\boldsymbol{v}_t^*) = (\boldsymbol{u}_1^*,\boldsymbol{v}_1^*)t+(\boldsymbol{u}_2^*,\boldsymbol{v}_2^*)(1-t) \tag{B.4}$$

is also a minimizer of (B.3) for all $t\in[0,1]$, with the same minimum value $\boldsymbol{u}_t^*+\boldsymbol{v}_t^*=\boldsymbol{u}_1^*+\boldsymbol{v}_1^*=\boldsymbol{u}_2^*+\boldsymbol{v}_2^*$. Generalizing this argument to the case of more than two extremal solutions, we conclude that all minimizers are global, with the same minimum value, and they live on the simplex where $\boldsymbol{e}^T(\boldsymbol{u}+\boldsymbol{v})=\boldsymbol{e}^T(\boldsymbol{u}_1+\boldsymbol{v}_1)$. Therefore, nonuniqueness requires that that this simplex has a nontrivial intersection with the feasible set where $\boldsymbol{\Phi}\boldsymbol{u}-\boldsymbol{\Phi}\boldsymbol{v}=\boldsymbol{y}$ with $\boldsymbol{u}\geq\boldsymbol{0}$, $\boldsymbol{v}\geq\boldsymbol{0}$. We argue that, generically, this will not be the case, i.e. the intersection will be trivial, and the extremal solution unique. In particular, since in our case we are in fact interested in the problem (B.1), we can always perturb slightly the discretization into H atoms of γ to guarantee that the extremal solution is unique. Since this is true no matter how large H is, and any Radon measure can be approached to arbitrary precision using such discretization, we conclude that the minimizer of (B.1) should be unique as well, with at most n atoms.

C Proof of Proposition 1

In this section, we provide the formal statement and proof of Proposition 1. Let us recall the general form of the predictor for both lazy and feature regimes in $d=2$. From Eq. 3.6,

$$f^n(x) = \sum_{j=1}^n g_j\tilde{\varphi}(x-x_j) = \int \frac{dy}{2\pi} g^n(y)\tilde{\varphi}(x-y). \tag{C.1}$$

where n is the number of training points for the lazy regime and the number of atoms for the feature regime and, for $x\in(-\pi,\pi]$,

$$\tilde{\varphi}(x) = \begin{cases} \max\{0,\cos(x)\} & \text{(feature regime)}, \\ \dfrac{2(\pi-|x|)\cos(x)+\sin(|x|)}{2\pi} & \text{(lazy regime, NTK)}, \\ \dfrac{(\pi-|x|)\cos(x)+\sin(|x|)}{2\pi} & \text{(lazy regime, RFK)}. \end{cases} \tag{C.2}$$

All these functions $\tilde{\varphi}$ have jump discontinuities on some derivative: the first for feature and NTK, the third for RFK. If the l-th derivative has jump discontinuities, the $l+1$-th only exists in a distributional sense and it can be generically written as a sum of a regular function and a sequence of Dirac masses located at the discontinuities. With m denoting the number of such discontinuities and $\{x_j\}_j$ their locations, $f^{(l)}$ denoting the l-th derivative of f, for some $c_j \in \mathbb{R}$,

$$f^{(l+1)}(x) = f_r^{(l+1)}(x) + \sum_{j=1}^{m} c_j \delta(x - x_j), \tag{C.3}$$

where f_r denotes the *regular* part of f.

Proposition 2. *Consider a random target function f^* satisfying Eq. 2.1 and the predictor f^n obtained by training a one-hidden-layer ReLU network on n samples $(x_i, f^*(x_i))$ in the feature or in the lazy regime (Eq. C.1). Then, with $\widehat{f}(k)$ denoting the Fourier transform of $f(x)$, one has*

$$\lim_{|k|\to\infty} \lim_{n\to\infty} \frac{\widehat{(f^n)''_r}(k)}{\widehat{f^*}(k)} = c, \tag{C.4}$$

where c is a constant (different for every regime). This result implies that as $n \to \infty$, $(f^n)''(x)$ converges to a function having finite second moment, i.e.

$$\begin{aligned}\lim_{n\to\infty} \mathbb{E}_{f^*} \left[(f^n)''_r(x)\right]^2 &= \lim_{n\to\infty} \mathbb{E}_{f^*} \left[\int dx \left((f^n)''_r\right)^2(x)\right] \\ &= \lim_{n\to\infty} \mathbb{E}_{f^*} \left[\sum_k \widehat{(f^n)''_r}^2(k)\right] = \textit{const.} < \infty,\end{aligned} \tag{C.5}$$

using the fact that $\mathbb{E}_{f^*}[(f^n)''_r(x)]^2$ does not depend on x and $\mathbb{E}_{f^*}[\sum_k \widehat{(f^*)}^2(k)] = const.$

Proof: Because our target functions are random fields that are in L_2 with probability one, and the RKHS of our kernels are dense in that space, we know that the test error vanishes as $n \to \infty$ [59]. As a result

$$f^*(x) = \lim_{n\to\infty} f^n(x) = \lim_{n\to\infty} \int \frac{dy}{2\pi} g^n(y) \tilde{\varphi}(x-y). \tag{C.6}$$

Consider first the feature regime and the NTK lazy regime. In both cases $\tilde{\varphi}$ has two jump discontinuities in the first derivative, located at $x = 0, \pi$ for the NTK and at $x = \pm\pi/2$, therefore we can write the second derivative as the sum of a regular function and two Dirac masses,

$$\begin{aligned}(\tilde{\varphi}^{\text{FEATURE}})'' &= -\max\{0, \cos(x)\} + \delta(x - \pi/2) + \delta(x + \pi/2), \\ (\tilde{\varphi}^{\text{NTK}})'' &= \frac{-2(\pi - |x|)\cos(x) + 3\sin(|x|)}{2\pi} - \frac{1}{2\pi}\delta(x) + \frac{1}{2\pi}\delta(x - \pi).\end{aligned} \tag{C.7}$$

As a result, the second derivative of the predictor can be written as the sum of a regular part $(f^n)''_r$ and a sequence of $2n$ Dirac masses. After subtracting the Dirac masses, both sides of Eq. C.1 can be differentiated twice and yield

$$(f^n)''_r(x) = \int \frac{dy}{2\pi} g^n(y) \tilde{\varphi}''_r(x - y). \tag{C.8}$$

Hence in the Fourier representation we have

$$\widehat{(f^n)''_r}(k) = \widehat{g^n}(k)(-k^2 \widehat{\tilde{\varphi}}_r(k)) \tag{C.9}$$

where we defined

$$\widehat{\tilde{\varphi}}(k) = \int_{-\pi}^{\pi} \frac{dx}{\sqrt{2\pi}} e^{ikx} \tilde{\varphi}(x), \qquad \widehat{\tilde{\varphi}_r}(k) = \int_{-\pi}^{\pi} \frac{dx}{\sqrt{2\pi}} e^{ikx} \tilde{\varphi}_r(x). \tag{C.10}$$

and used $\widehat{\tilde{\varphi}''_r}(k) = -k^2 \widehat{\tilde{\varphi}_r}(k)$. By universal approximation we have

$$\widehat{f^*}(k) = \int_{-\pi}^{\pi} \frac{dx}{\sqrt{2\pi}} e^{ikx} f^*(x) = \lim_{n\to\infty} \widehat{g^n}(k) \widehat{\tilde{\varphi}}(k) \qquad \Rightarrow \qquad \lim_{n\to\infty} \widehat{g^n}(k) = \frac{\widehat{f^*}(k)}{\widehat{\tilde{\varphi}}(k)}. \tag{C.11}$$

As a result by combining Eq. C.9 and Eq. C.11 we deduce

$$\lim_{n\to\infty} \widehat{(f^n)''_r}(k) = -\frac{k^2 \widehat{\tilde{\varphi}}_r(k)}{\widehat{\tilde{\varphi}}(k)} \widehat{f^*}(k). \tag{C.12}$$

To complete the proof using this result it remains to estimate the scaling of $\widehat{\tilde{\varphi}}_r(k)$ and $\widehat{\tilde{\varphi}}(k)$ in the large $|k|$ limit.

For the feature regime, a direct calculation shows that $\tilde{\varphi}_r'' = -\tilde{\varphi}$, implying that $\widehat{\tilde{\varphi}}_r(k) = -\widehat{\tilde{\varphi}}(k)$. This proves that Eq. C.4 is satisfied with $c = -1$.

For the NTK lazy regime $\tilde{\varphi}_r''$ and $-\tilde{\varphi}$ are different but they have similar singular expansions near $x=0$ and π. Therefore their Fourier coefficients display the same asymptotic decay. More specifically, with $t = \cos(x)$ (or $x = \arccos(t)$), so that $\tilde{\varphi}(x) = \varphi(t)$, one has

$$\begin{cases} \varphi^{\rm NTK}(t) = t - \dfrac{1}{\sqrt{2}\pi}(1-t)^{1/2} + O\left((1-t)^{3/2}\right) \text{ near } t = +1; \\ \varphi^{\rm NTK}(t) = -\dfrac{1}{\sqrt{2}\pi}(-1+t)^{1/2} + O\left((-1+t)^{3/2}\right) \text{ near } t = -1, \end{cases} \tag{C.13}$$

and

$$\begin{cases} (\varphi^{\rm NTK})_r''(t) = -t + \dfrac{5}{\sqrt{2}\pi}(1-t)^{1/2} + O\left((1-t)^{3/2}\right) \text{ near } t = +1; \\ (\varphi^{\rm NTK})_r''(t) = +\dfrac{5}{\sqrt{2}\pi}(-1+t)^{1/2} + O\left((-1+t)^{3/2}\right) \text{ near } t = -1. \end{cases} \tag{C.14}$$

Therefore, due to Eq. A.17, Eq. C.4 is satisfied with $c = -5$. The same procedure can be applied to the RFK lazy regime, with the exception that it is the fourth derivative of $\tilde{\varphi}^{\rm RFK}$ which can be written as a regular part plus Dirac masses, but one can still obtain the Fourier coefficients of the second derivative's regular part by dividing those of the fourth derivative's regular part by k^2.

D Asymptotics of generalization in $d=2$

In this section we compute the decay of generalization error $\bar{\epsilon}$ with the number of samples n in the following 2-dimensional setting:

$$f^n(x) = \sum_{j=1}^{n} g_j \tilde{\varphi}(x - x_j), \tag{D.1}$$

where the x_j's are the training points (like in the NTK case) and φ has a single discontinuity on the first derivative in 0.

Let us order the training points clockwise on the ring, such that $x_1 = 0$ and $x_{i+1} > x_i$ for all $i = 1, \dots, n$, with $x_{n+1} := 2\pi$. On each of the x_i the predictor coincides with the target,

$$f^n(x_i) = f^*(x_i) \quad \forall\, i = 1, \dots, n. \tag{D.2}$$

For large enough n, the difference $x_{i+1} - x_i$ is small enough such that, within (x_i, x_{i+1}), $f^n(x)$ can be replaced with its Taylor series expansion up to the second order. In practice, the predictors appear like the cable of a suspension bridge with the pillars located on the training points. In particular, we can consider an expansion around $x_i^+ := x_i + \epsilon$ for any $\epsilon > 0$ and then let $\epsilon \to 0$ from above:

$$f^n(x) = f^n(x_i^+) + (x - x_i^+) f^{n\prime}(x_i^+) + \frac{(x - x_i^+)^2}{2} (f^n)''(x_i^+) + \mathcal{O}\left((x - x_i^+)^3\right). \tag{D.3}$$

By differentiability of f^n in (x_i, x_{i+1}) the second derivative can be computed at any point inside (x_i, x_{i+1}) without changing the order of approximation in Eq. D.3, in particular we can replace $(f^n)''(x_i^+)$ with c_i, the mean curvature of f^n in (x_i, x_{i+1}). Moreover, as $\epsilon \to 0$, $f^n(x_i^+) \to f^*(x_i)$ and $f^n(x_{i+1}^-) \to f^*(x_{i+1})$. By introducing the limiting slope $m_i^+ := \lim_{x \to 0^+} f^{n\prime}(x_i + x)$, we can write

$$f^n(x) = f^*(x_i) + (x - x_i) m_i^+ + \frac{(x - x_i)^2}{2} c_i + O\left((x - x_i^+)^3\right) \tag{D.4}$$

Computing Eq. D.4 at $x = x_{i+1}$ yields a closed form for the limiting slope m_i^+ as a function of the mean curvature c_i, the interval length $\delta_i := (x_{i+1} - x_i)$ and $\Delta f_i := f^*(x_{i+1}) - f^*(x_i)$. Specifically,

$$m_i^+ = \frac{\Delta f_i}{\delta_i} - \frac{\delta_i}{2} c_i. \tag{D.5}$$

The generalization error can then be split into contributions from all the intervals. If $\nu_t > 2$, A Taylor expansion leads to:

$$\begin{aligned}
\epsilon(n) &= \int_0^{2\pi} \frac{dx}{2\pi} \left(f^n(x) - f^*(x)\right)^2 \\
&= \sum_{i=1}^n \int_{x_i}^{x_{i+1}} \frac{dx}{2\pi} \left[(x-x_i)\left(m_i^+ - (f^*)'(x_i)\right) + \frac{(x-x_i)^2}{2}\left(c_i - (f^*)''(x_i)\right) + o\left((x-x_i^+)^2\right)\right]^2 \\
&= \sum_{i=1}^n \int_0^{\delta_i} \frac{d\delta}{2\pi} \left[\delta\left(m_i^+ - (f^*)'(x_i)\right) + \frac{\delta^2}{2}\left(c_i - (f^*)''(x_i)\right) + o\left(\delta^2\right)\right]^2 \\
&= \sum_{i=1}^n \frac{1}{2\pi} \left[\frac{\delta_i^3}{3}\left(m_i^+ - (f^*)'(x_i)\right)^2 + \frac{\delta_i^5}{20}\left(c_i - (f^*)''(x_i)\right)^2 \right. \\
&\qquad\qquad \left. + \frac{\delta_i^4}{4}\left(m_i^+ - (f^*)'(x_i)\right)\left(c_i - (f^*)''(x_i)\right) + o(\delta_i^5)\right].
\end{aligned} \tag{D.6}$$

In addition, as $\Delta f_i = (f^*)'(x_i)\delta_i + (f^*)''(x_i)\delta_i^2/2 + O(\delta_i^3)$,

$$m_i^+ - (f^*)'(x_i) = \frac{\delta_i}{2}\left((f^*)''(x_i) - c_i\right) + o(\delta_i)^2, \tag{D.7}$$

thus

$$\epsilon(n) = \frac{1}{2\pi}\sum_{i=1}^n \left[\frac{\delta_i^5}{120}\left(c_i - (f^*)''(x_i)\right)^2 + o(\delta_i^5)\right]. \tag{D.8}$$

implying:

$$\bar{\epsilon}(n) = \frac{n^{-4}\left(n^{-1}\sum_{i=1}^n (n\delta_i)^5\right)}{240\pi} \lim_{n\to\infty} \int \mathbb{E}_{f^*}\left[\left((f^n)''(x) - (f^*)''(x)\right)^2\right] dx + o(n^{-4}) \sim \frac{1}{n^4} \tag{D.9}$$

where we used that *(i)* the integral converges to some finite value, due to proposition 2. From App. C, this integral can be estimated as $\sum_k \mathbb{E}_{f^*}\left[\left(cf^*(k) - k^2 f^*(k)\right)^2\right]$, that indeed converges for $\nu_t > 2$. *(ii)* $\left(n^{-1}\sum_{i=1}^n (n\delta_i)^5\right)$ has a deterministic limit for large n. It is clear for the lazy regime since the distance between adjacent singularities δ_i follows an exponential distribution of mean $\sim \frac{1}{n}$. We expect this result to be also true for the feature regime in our set-up. Indeed, in the limit $n \to \infty$, the predictor approaches a parabola between singular points, which generically cannot fit more than three random points. There must thus be a singularity at least every two data-points with a probability approaching unity as $n \to \infty$, which implies that $\left(n^{-1}\sum_{i=1}^n (n\delta_i)^5\right)$ converges to a constant for large n.

Finally, for $\nu_t < 2$, the same decomposition in intervals applies, but a Taylor expansion to second order does not hold. The error is then dominated by the fluctuations of f^* on the scale of the intervals, as indicated in the main text.

E Asymptotic of generalization via the spectral bias ansatz

According to the spectral bias ansatz, the first n modes of the predictor $f_{k,\ell}^n$ coincide with the modes of the target function $f_{k,\ell}^*$. Therefore, the asymptotic scaling of the error with n is entirely controlled by the remaining modes,

$$\epsilon(n) \sim \sum_{k\geq k_c}\sum_{\ell=1}^{\mathcal{N}_{k,d}} \left(f_{k,\ell}^n - f_{k,\ell}^*\right)^2 \quad \text{with} \quad \sum_{k\leq k_c} \mathcal{N}_{k,d} \sim n. \tag{E.1}$$

Since $\mathcal{N}_{k,d} \sim k^{d-2}$ for $k \gg 1$, one has that, for large n, $k_c \sim n^{\frac{1}{d-1}}$. After averaging the error over target functions we get

$$\bar{\epsilon}(n) \sim \sum_{k\geq k_c}\sum_{\ell=1}^{\mathcal{N}_{k,d}} \left\{\mathbb{E}_{f^*}\left[\left(f_{k,\ell}^n\right)^2\right] + \mathbb{E}_{f^*}\left[\left(f_{k,\ell}^*\right)^2\right] - 2\mathbb{E}_{f^*}\left[\left(f_{k,\ell}^n f_{k,\ell}^*\right)\right]\right\}. \tag{E.2}$$

Let us recall that, with the predictor having the general form in Eq. 3.2, then

$$f^n_{k,\ell} = g^n_{k,\ell}\varphi_k \quad \text{with} \quad g^n_{k,\ell} = \sum_{j=1}^{n} g_j Y_{k,\ell}(\boldsymbol{y}_j), \tag{E.3}$$

where the $\boldsymbol{y}_j$'s denote the training points for the lazy regime and the neuron features for the feature regime. For $k \ll k_c$, where $f^n_{k,\ell} = f^*_{k,\ell}$, $g^n_{k,\ell} = f^*_{k,\ell}/\varphi_k$. For $k \gg k_c$, due to the highly oscillating nature of $Y_{k,\ell}$, the factors $Y_{k,\ell}(\boldsymbol{y}_j)$ are essentially decorrelated random numbers with zero mean and finite variance, since the values of $(Y_{k,\ell}(\boldsymbol{y}_j))^2$ are limited by the addition theorem Eq. A.5. Let us denote the variance with σ_Y. By the central limit theorem, $g^n_{k,\ell}$ converges to a Gaussian random variable with zero mean and finite variance $\sigma_Y^2 \sum_{j=1}^n g_j^2$. As a result,

$$\begin{aligned} \bar{\epsilon}(n) &\sim \sum_{k \geq k_c} \sum_{\ell=1}^{\mathcal{N}_{k,d}} \left\{ \left(\sum_{j=1}^{n} g_j^2 \right) \varphi_k^2 + \mathbb{E}_{f^*}\left[\left(f^*_{k,\ell} \right)^2 \right] \right\} \\ &= \left(\sum_{j=1}^{n} g_j^2 \right) \sum_{k \geq k_c} \mathcal{N}_{k,d}\varphi_k^2 + \sum_{k \geq k_c} \mathcal{N}_{k,d} c_k, \end{aligned} \tag{E.4}$$

where we have used the definition of f^* (Eq. 2.1) to set the expectation of $(f^*_{k,\ell})^2$ to c_k.

Large ν_t case When f^* is smooth enough the error is controlled by the predictor term proportional to $\sum_{j=1}^n g_j^2$. More specifically, if

$$\sum_{k \geq 0} \sum_{\ell=1}^{\mathcal{N}_{k,d}} \frac{c_k}{\varphi_k^2} < +\infty, \tag{E.5}$$

then the function $g^n(\boldsymbol{x})$ converges to the square-summable function $g^*(\boldsymbol{x})$ such that $f^*(\boldsymbol{x}) = \int g^*(\boldsymbol{y})\varphi(\boldsymbol{x} \cdot \boldsymbol{y})\, d\tau(\boldsymbol{y})$. With $c_k \sim k^{-2\nu_t-(d-1)}$ and $\mathcal{N}_{k,d} \sim k^{d-2}$, in the lazy regime $\varphi_k \sim k^{-(d-1)-2\nu}$ Eq. E.5 is satisfied when $2\nu_t > 2(d-1) + 4\nu$ ($\nu = 1/2$ for the NTK and $3/2$ for the RFK). In the feature regime $\varphi_k \sim k^{-(d-1)/2-3/2}$, Eq. E.5 is satisfied when $2\nu_t > (d-1) + 3$. If $g^n(\boldsymbol{x})$ converges to a square-summable function, then

$$\sum_{j=1}^{n} g_j^2 = \frac{1}{n} \int g^n(\boldsymbol{x})^2 \, d\tau(\boldsymbol{x}) + o(n^{-1}) = \frac{1}{n} \sum_{k \geq 0} \mathcal{N}_{k,d} \frac{c_k}{\varphi_k^2} + o(n^{-1}), \tag{E.6}$$

which is proportional to n^{-1}. In addition, since $\mathcal{N}_{k,d} \sim k^{d-2}$ and $k_c \sim n^{\frac{1}{d-1}}$, one has

$$n^{-1} \sum_{k \geq k_c} \mathcal{N}_{k,d}\varphi_k \sim \begin{cases} n^{-1} k^{d-1} k^{-2(d-1)-4\nu} \Big|_{k=n^{\frac{1}{d-1}}} \sim n^{-2-\frac{4\nu}{d-1}} \text{ (Lazy)}, \\ n^{-1} k^{d-1} k^{-(d-1)-3} \Big|_{k=n^{\frac{1}{d-1}}} \sim n^{-1-\frac{3}{d-1}} \text{ (Feature)}, \end{cases} \tag{E.7}$$

and

$$\sum_{k \geq k_c} \mathcal{N}_{k,d} c_k \sim k^{d-1} k^{-2\nu_t-(d-1)} \Big|_{k=n^{\frac{1}{d-1}}} \sim n^{-\frac{2\nu_t}{d-1}}. \tag{E.8}$$

Hence, if ν_t is large enough so that Eq. E.5 is satisfied, the asymptotic decay of the error is given by Eq. E.7.

Small ν_t case If Eq. E.7 does not hold then $g^n(\boldsymbol{x})$ is not square-summable in the limit $n \to \infty$. However, for large but finite n only the modes up to the k_c-th are correctly reconstructed, therefore

$$\sum_{j=1}^{n} g_j^2 \sim \frac{1}{n} \sum_{k \leq k_c} \mathcal{N}_{k,d} \frac{c_k}{\varphi_k^2} \sim \begin{cases} n^{-1} k^{-2\nu_t} k^{2(d-1)+4\nu} \Big|_{k=n^{\frac{1}{d-1}}} \sim n^{-\frac{2\nu_t}{d-1}} n^{1+\frac{4\nu}{d-1}} \text{ (Lazy)}, \\ n^{-1} k^{-2\nu_t} k^{(d-1)+3} \Big|_{k=n^{\frac{1}{d-1}}} \sim n^{-\frac{2\nu_t}{d-1}} n^{\frac{3}{d-1}} \text{ (Feature)}, \end{cases} \tag{E.9}$$

Both for feature and lazy, multiplying the term above by $\sum_{k \geq k_c} \mathcal{N}_{k,d}\varphi_k$ from Eq. E.7 yields $\sim n^{-2\nu_t/(d-1)}$. This is also the scaling of the target function term Eq. E.8, implying that for small ν_t one has

$$\bar{\epsilon}(n) \sim n^{-\frac{2\nu_t}{d-1}} \tag{E.10}$$

both in the feature and in the lazy regimes.

F Spectral bias via the replica calculation

Due to the equivalence with kernel methods, the asymptotic decay of the test error in the lazy regime can be computed with the formalism of [45], which also provides a non-rigorous justification for the spectral bias ansatz. By ranking the eigenvalues from the biggest to the smallest, such that φ_ρ denotes the ρ-th eigenvalue and denoting with c_ρ the variance of the projections of the target onto the ρ-th eigenfunction, one has

$$\epsilon(n) = \sum_\rho \epsilon_\rho(n), \quad \epsilon_\rho(n) = \frac{\kappa(n)^2}{(\varphi_\rho + \kappa(n))^2} c_\rho, \quad \kappa(n) = \frac{1}{n} \sum_\rho \frac{\varphi_\rho \kappa(n)}{\varphi_\rho + \kappa(n)}. \tag{F.1}$$

It is convenient to introduce the eigenvalue density,

$$\mathcal{D}(\varphi) := \sum_{k\geq 0} \sum_{l=1}^{\mathcal{N}_{k,d}} \delta(\varphi - \varphi_k) = \sum_{k\geq 0} \mathcal{N}_{k,d} \delta(\varphi - \varphi_k) \sim \int_0^\infty k^{d-2} \delta(\varphi - k^{-(d-1)-2\nu}) \text{ for } k \gg 1. \tag{F.2}$$

After changing variables in the delta function, one finds

$$\mathcal{D}(\varphi) \sim \varphi^{-\frac{2(d-1)+2\nu}{(d-1)+2\nu}} \text{ for } \varphi \ll 1. \tag{F.3}$$

This can be used for inferring the asymptotics of $\kappa(n)$,

$$\begin{aligned} \kappa(n) &= \frac{1}{n} \sum_\rho \frac{\varphi_\rho \kappa(n)}{\varphi_\rho + \kappa(n)} \sim \frac{1}{n} \int d\varphi\, \mathcal{D}(\varphi) \frac{\varphi \kappa(n)}{\varphi + \kappa(n)} \\ &\sim \frac{1}{n} \int_0^{\kappa(n)} d\varphi\, \mathcal{D}(\varphi) \varphi + \frac{\kappa(n)}{n} \int_{\kappa(n)}^{\varphi_0} d\varphi\, \mathcal{D}(\varphi) \\ &\sim \frac{1}{n} \kappa(n)^{1-\frac{(d-1)}{(d-1)+2\nu}} \Rightarrow \kappa(n) \sim n^{-1-\frac{2\nu}{d-1}}. \end{aligned} \tag{F.4}$$

Once the scaling of $\kappa(n)$ has been determined, the modal contributions to the error can be split according to whether $\varphi_\rho \ll \kappa(n)$ or $\varphi_\rho \gg \kappa(n)$. The scaling of φ_ρ with the rank ρ is determined self-consistently,

$$\rho \sim \int_{\varphi_\rho}^{\varphi_1} d\varphi\, \mathcal{D}(\varphi) \sim \varphi_\rho^{-\frac{d-1}{(d-1)+2\nu}} \Rightarrow \varphi_\rho \sim \rho^{-1-\frac{2\nu}{d-1}} \Rightarrow \varphi_\rho \gg (\ll) \kappa(n) \Leftrightarrow \rho \ll (\gg) n. \tag{F.5}$$

Therefore

$$\epsilon(n) \sim \kappa(n)^2 \sum_{\rho \ll n} \frac{c_\rho}{\varphi_\rho^2} + \sum_{\rho \gg n} c_\rho. \tag{F.6}$$

Notice that $\kappa(n)^2$ scales as $n^{-1} \sum_{k\geq k_c} \mathcal{N}_{k,s} \varphi_k$ in Eq. E.7, whereas $\sum_{\rho \ll n} c_\rho / \varphi_\rho^2$ corresponds to $n \sum_j g_j^2$ in Eq. E.9, so that the first term on the right-hand side of Eq. F.6 matches that of Eq. E.4. The same matching is found for the second term on the right-hand side of Eq. F.6, so that the replica calculation justifies the spectral bias ansatz.

G Training wide neural networks: does gradient descent (GD) find the minimal-norm solution?

In the main text we provided predictions for the asymptotics of the test error of the minimal norm solution that fits all the training data. Does the prediction hold when solution of Eq. 2.5 and Eq. 2.13 is approximately found by GD? More specifically, is the solution found by GD the minimal-norm one?

Feature Learning We answer these questions by performing full-batch gradient descent in two settings (further details about the trainings are provided in the code repository, `experiments.md` file),

1. **Min-L1.** Here we update weights and features of Eq. 2.3, with $\xi = 0$, by following the negative gradient of

$$\mathcal{L}_{\text{Min-L1}} = \frac{1}{2n}\sum_{i=1}^{n}\left(f^*(\boldsymbol{x}_i) - f(\boldsymbol{x}_i)\right)^2 + \frac{\lambda}{H}\sum_{h=1}^{H}|w_h|, \tag{G.1}$$

with $\lambda \to 0^+$. The weights w_h are initialized to zero and the features are initialized uniformly and constrained to be on the unit sphere.

2. α**-trick.** Following [8], here we minimize

$$\mathcal{L}_{\alpha\text{-trick}} = \frac{1}{2n\alpha}\sum_{i=1}^{n}\left(f^*(\boldsymbol{x}_i) - \alpha f(\boldsymbol{x}_i)\right)^2, \tag{G.2}$$

with $\alpha \to 0$. This trick allows to be far from the lazy regime by forcing the weights to evolve to $\mathcal{O}(1/\alpha)$, when fitting a target of order 1.

In both cases, the solution found by GD is sparse, in the sense that is supported on a finite number of neurons – in other words, the measure $\gamma(\boldsymbol{\theta})$ becomes atomic, satisfying Assumption 1. Furthermore, we find that

1. For **Min-L1**, the generalization error prediction holds (Fig. 4 and Fig. G.1) as the the minimal norm solution if effectively recovered, see Fig. G.2. Such clean results in terms of features position are difficult to achieve for large n because the training dynamics becomes very slow and reaching convergence becomes computationally infeasible. Still, we observe the test error to plateau and reach its infinite-time limit much earlier than the parameters, which allows for the scaling predictions to hold.
2. α**-trick**, however, does not recover the minimal-norm solution, Fig. G.2. Still, the solution found is of the type (2.7) as it is sparse and supported on a number of atoms that scales linearly with n, Fig. G.3, left. For this reason, we find that our predictions for the generalization error hold also in this case, see Fig. G.3, right.

Lazy Learning In this case, the correspondence between the solution found by gradient descent and the minimal-norm one is well established [9]. Therefore, numerical experiments are performed here via kernel regression and the analytical NTK Eq. A.19: given a dataset $\{\boldsymbol{x}_i, y_i = f^*(\boldsymbol{x}_i)\}_{i=1}^n$, we define the gram matrix $\mathbf{K} \in \mathbb{R}^{n\times n}$ with elements $\mathbf{K}_{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and the vector of target labels $\boldsymbol{y} = [y_1, y_2, \dots, y_n]$. The q_i's in Eq. 2.9 can be easily recovered by solving the linear system

$$\boldsymbol{y} = \tfrac{1}{n}\mathbf{K}\boldsymbol{q}. \tag{G.3}$$

Experiments Numerical experiments are run with PyTorch on GPUs NVIDIA V100 (university internal cluster). Details for reproducing experiments are provided in the code repository, `experiments.md` file. Individual trainings are run in 1 minute to 1 hour of wall time. We estimate a total of a thousand hours of computing time for running the preliminary and actual experiments present in this work.

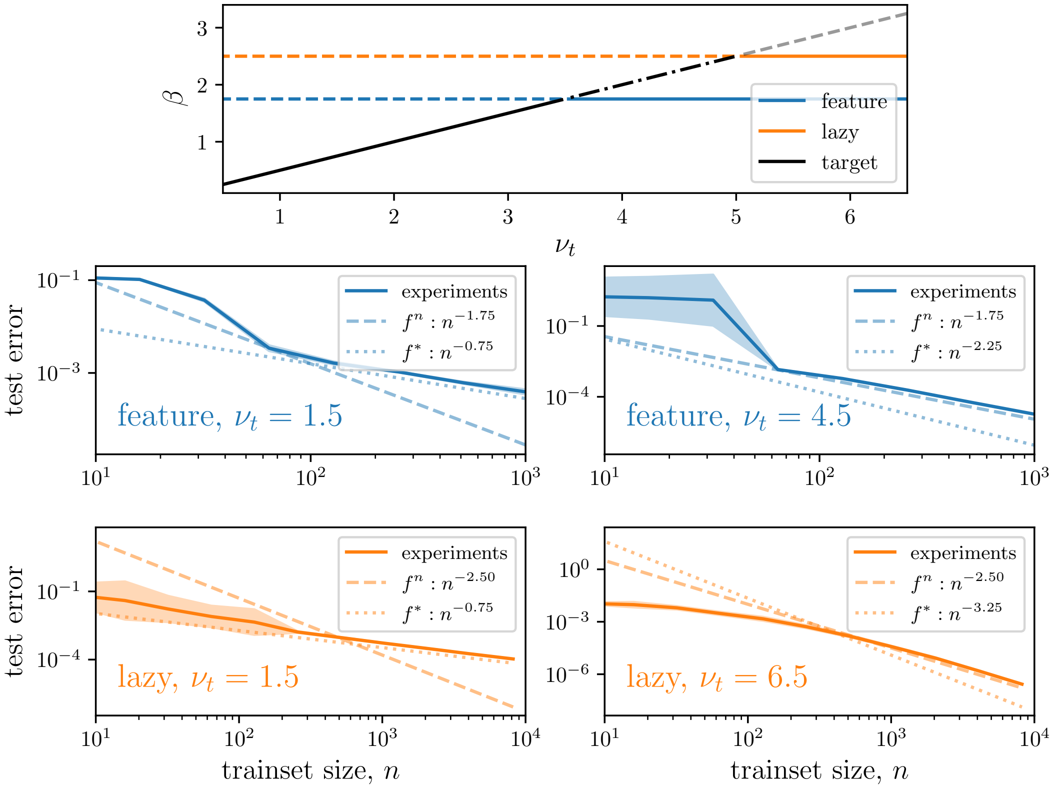


Figure G.1: **Gen. error decay vs. target smoothness and training regime.** Here, data-points are sampled uniformly from the spherical surface in $d = 5$ and the target function is an infinite-width FCN with activation function $\sigma(\cdot) = |\cdot|^{\nu_t - 1/2}$, corresponding to a Gaussian random process of smoothness ν_t. 1st row: gen. error decay exponent as a function of the target smoothness ν_t. The three curves correspond to the target contribution to the generalization error (black) and the predictor contribution in either feature (blue) or lazy (orange) regime. Full lines highlight the dominating contributions to the gen. error. 2nd row: agreement between predictions and experiments in the feature regime for a non-smooth (left) and smooth (right) target. In the first case, the error is dominated by the target f^*, in the second by the predictor f^n – predicted exponents β are indicated in the legends. 3rd row: analogous of the previous row for the lazy regime.

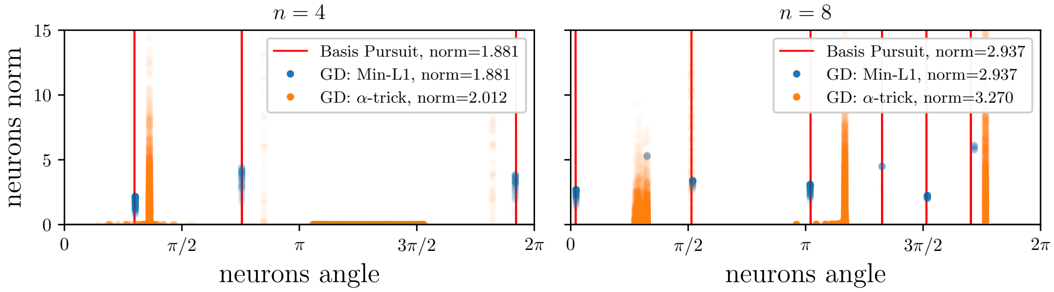


Figure G.2: **Comparing solutions.** Solutions to the spherically symmetric task in $d = 2$ for $n = 4$ (left) and $n = 8$ (right) training points. In red the minimal norm solution (Eq. 2.5) as found by Basis Pursuit [50]. Solutions found by GD in the Min-L1 and α-trick setting are respectively shown in blue and orange. Dots correspond to single neurons in the network. The x-axis reports their angular position while the y-axis reports their norm: $|w_h|\|\boldsymbol{\theta}_h\|_2$. The total norm of the solutions, $\frac{\alpha}{H}\sum_{h=1}^{H}|w_h|\|\boldsymbol{\theta}_h\|_2$, is indicated in the legend.

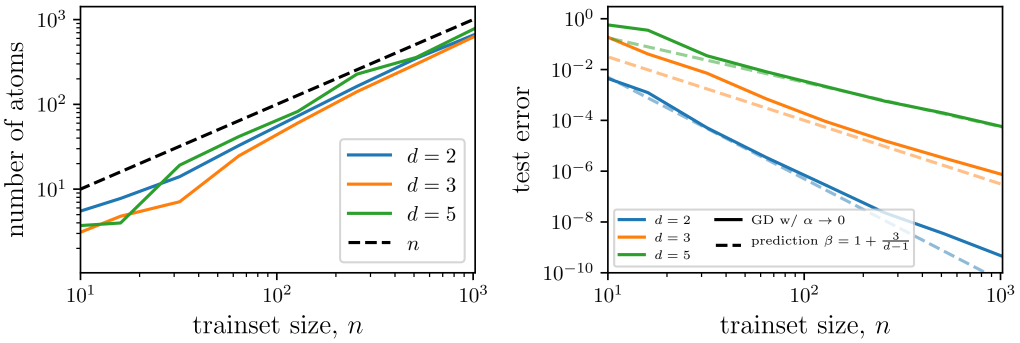


Figure G.3: **Solution found by the α-trick.** We consider here the case of approximating the constant target function on $\mathbb{S}^{d-1}$ with an FCN. Training is performed starting from small initialization through the α-trick. Left: Number of atoms n_A as a function of the number of training points n. Neurons that are active on the same subset of the training set are grouped together and we consider each group a distinct atom for the counting. Right: Generalization error in the same setting (full), together with the theoretical predictions (dashed). Different colors correspond to different input dimensions. The case of $d = 2$ and large n suffers from the same finite time effects discussed in Fig. 4. Results are averaged over 10 different initializations of the networks and datasets.

H Sensitivity of the predictor to transformations other than diffeomorphisms

This section reports experiments to integrate the discussion of section 5. In particular, we: *(i)* show that the lazy regime predictor is less sensitive to image translations than the feature regime one (as is the case for deformations, from Fig. 6); *(ii)* provide evidence of the positive effects of learning features in image classifications, namely becoming invariant to pixels at the border of images which are unrelated to the task.

To prove the above points we consider, as in Fig. 6, the relative sensitivity of the predictors of lazy and feature regime with respect to global translations for point *(i)* and corruption of the boundary pixels for point *(ii)*. The relative sensitivity to translations is obtained from Eq. 5.1 after replacing the transformation τ with a one-pixel translation of the image in a random direction. For the relative sensitivity to boundary corruption, the transformation consists in adding zero-mean and unit-variance Gaussian numbers to the boundary pixels. Both relative sensitivities are plotted in Fig. H.1, with translations on the left and boundary pixels corruption on the right.

In section 5 we then argue that differences in performance between the two training regimes can be explained by gaps in sensitivities with respect to input transformations that do not change the label. For *(i)*, the gap is similar to the one observed for diffeomorphisms (Fig. 6). Still, the space of translations has negligible size with respect to input space, hence we expect the diffeomorphisms to have a more prominent effect. In case *(ii)*, the feature regime is less sensitive with respect to irrelevant pixels corruption and this would give it an advantage over the lazy regime. The fact that the performance difference is in favor of the lazy regime instead, means that these transformations only play a minor role.

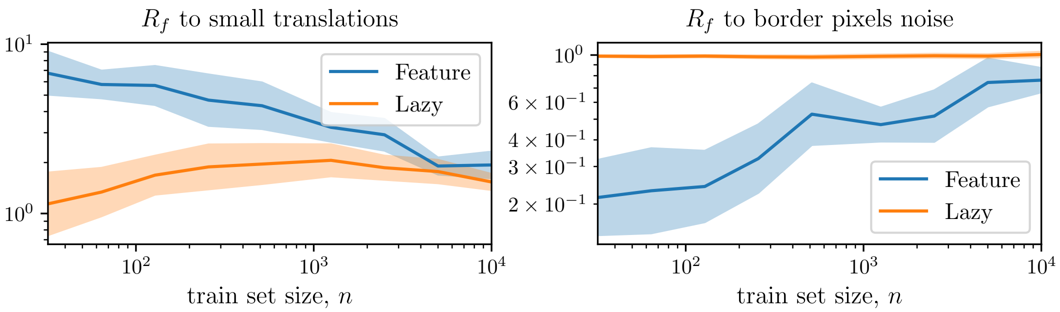


Figure H.1: **Sensitivity to input transformations vs number of training points.** Relative sensitivity of the predictor to (left) random 1-pixel translations and (right) white noise added at the boundary of the input images, in the two regimes, for varying number of training points n and when training on FashionMNIST. Smaller values correspond to a smoother predictor, on average. Results are computed using the same predictors as in Fig. 1. Left: For small translations, the behavior is the same compared to applying diffeomorphisms. Right: The lazy regime does not distinguish between noise added at the boundary or on the whole image ($R_f = 1$), while the feature regime gets more insensitive to the former.

I Maximum-entropy model of diffeomorphisms

We briefly review here the maximum-entropy model of diffeomorphisms as introduced in [49].

An image can be thought of as a function $x(s)$ describing intensity in position $s = (u, v) \in [0, 1]^2$, where u and v are the horizontal and vertical (pixel) coordinates. Denote τx the image deformed by τ, i.e. $[\tau x](s) = x(s - \tau(s))$. [49] propose an ensemble of diffeomorphisms $\tau(s) = (\tau_u, \tau_v)$ with i.i.d. τ_u and τ_v defined as

$$\tau_u = \sum_{i,j \in \mathbb{N}^+} C_{ij} \sin(i\pi u) \sin(j\pi v) \tag{I.1}$$

where the C_{ij}'s are Gaussian variables of zero mean and variance $T/(i^2 + j^2)$ and T is a parameter controlling the deformation magnitude. Once τ is generated, pixels are displaced to random positions. See Fig. 5b for an example of such transformation.