How Much does Initialization Affect Generalization?

Sameera Ramasinghe [1] **Lachlan MacDonald** [2] **Moshiur Farazi** [3] **Hemanth Saratchandran** [2] **Simon Lucey** [2]

Abstract

Characterizing the remarkable generalization properties of over-parameterized neural networks remains an open problem. A growing body of recent literature shows that the bias of stochastic gradient descent (SGD) and architecture choice implicitly leads to better generalization. In this paper, we show on the contrary that, independently of architecture, SGD can itself be the cause of poor generalization if one does not ensure good initialization. Specifically, we prove that *any* differentiably parameterized model, trained under gradient flow, obeys a weak spectral bias law which states that sufficiently high frequencies train arbitrarily slowly. This implies that very high frequencies present at initialization will remain after training, and hamper generalization. Further, we empirically test the developed theoretical insights using practical, deep networks. Finally, we contrast our framework with that supplied by the *flat-minima* conjecture and show that Fourier analysis grants a more reliable framework for understanding the generalization of neural networks.

1. Introduction

Neural networks are often used in the over-parameterized regime, meaning their loss landscapes admit many global minima that achieve zero training error. However, finding such solutions is a non-convex, high-dimensional problem, which is typically intractable to solve analytically. Furthermore, each of these minima may have unique properties that can lead to varying generalization performance, making some solutions more preferred than others. Surprisingly, however, it is widely established that when neural networks are trained with gradient-based optimization techniques, they not only converge towards a global minimum, but also are biased towards solutions that exhibit good generalization even without explicit regularization. This mysterious behavior is flagged as the "implicit regularization" of neural networks and remains an open research problem.

To understand implicit regularization, numerous studies have considered simplified settings with restrictive assumptions such as linear networks (Jacot et al., 2018b; Soudry et al., 2018; Wei et al., 2019; Yun et al., 2020; Gunasekar et al., 2018b; Wu et al., 2019), shallow networks (Gunasekar et al., 2018a; Ji & Telgarsky, 2019; Ali et al., 2020), wide networks (Jacot et al., 2018a; Mei et al., 2019; Chizat & Bach, 2020; Oymak & Soltanolkotabi, 2019; Zhang et al., 2020a), vanishing initialization (Chizat et al., 2019; Gunasekar et al., 2017; Arora et al., 2019), or infinitesimal learning rates (Ma et al., 2018; Li et al., 2018; Ji & Telgarsky, 2018; Moroshko et al., 2020). Despite different assumptions, most of these works primarily focus on the effect of optimization procedure over the other factors and, at a high level, conclude that gradient-based optimizations guides neural networks toward max-margin solutions for separable data or minimize a notion of weight-norm in regression. While the aforementioned studies yield powerful insights, there remains a gap between theory and practice due to the restrictive assumptions presently necessary to prove quantitative results.

Our work is an attempt to help bridge this gap. To this end, we show substantial evidence that although the optimization procedure provides an important bias, initialization also plays a decisive role in determining the generalization of a neural network, and that this factor is at play across all architectures. In particular, we demonstrate that even with gradient-based optimization and a deep architecture – networks can converge to solutions with extremely poor generalization properties. We further demonstrate that this result depends on the Fourier spectrum at initialization. **It should be noted that our result is not a recapitulation of the well-known observation that bad initialization hampers the convergence of neural networks. Rather, we show that initializing networks such that they have higher energies for higher frequencies leads to solutions that achieve perfect training accuracy, yet succumb to inferior test accuracy**. We further reveal that this is a generic property that holds in both classification and regression

[1]Amazon, Australia [2]Australian Institute of Machine Learning, University of Adelaide, Adelaide SA, Australia [3]Machine Learning and Artificial Intelligence FSP, Data61-CSIRO. Correspondence to: Sameera Ramasinghe <sameera.ramasinghe@adelaide.edu.au>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

settings across various architectures.

The roots of our analysis extend to the "spectral bias" (also known as the frequency principle) of neural networks (Xu et al., 2019b; Rahaman et al., 2019). Spectral bias is the interesting phenomenon that neural networks tend to learn low frequencies faster, and consequently, tend to fit training data with low frequency functions. Significant progress has been made on understanding and quantifying this phenomenon (Rahaman et al., 2019; Xu, 2018; Xu et al., 2019b; Luo et al., 2019; 2020; Zhang et al., 2019; 2020a), however research up until this point has made assumptions on architecture (such as large width, limited depth, limited choice of activations, and chain-only architectures) which do not hold for many practical models. As has been noted in these previous works, the spectral bias has deep implications for generalization and its relationship to initialization. We contend that the provision of a more *general* theoretical and empirical analysis of spectral bias, one which applies even to practical models widely in use, will thus be of great value to the machine learning community..

The central objective of this paper is the provision of such a general analysis. To this end, we utilize recently popularized implicit neural networks (Tancik et al., 2020; Ramasinghe & Lucey, 2021) (also referred to as coordinate-based networks) as an initial test-bed. Implicit neural networks are architecturally modified fully-connected networks – using non traditional activations such as Gaussians/sinusoids or positional embedding layers – that can learn high-frequency functions rapidly. In particular, we first demonstrate that implicit neural networks do *not* always converge to smooth solutions, contradicting mainstream expectations. In resolving this surprising observation, first, we invoke a *compact data manifold hypothesis* to show that a weak form of spectral bias (namely that *sufficiently high frequencies train arbitrarily slowly*) is both architecture- and loss-agnostic in a general sense. The term "sufficiently high" is architecture dependent; the aim of this work is not to provide precise quantifications of the spectral bias over a subset of architectures, but is instead to present a more general result. Our qualitative theorem applies to *any* differentiably parameterized model, and our experiments suggest that for common neural networks this spectral decay is present even at quite low frequencies. With this in hand, we affirm that the poor generalization of implicit neural networks is linked to the presence of high frequencies at initialization which, due to the aforementioned weak spectral bias, tend to remain unchanged during training. Similarly, we further show that the implicit regularization of neural networks requires an initial spectrum that is biased towards lower frequencies. We postulate that the remarkable generalization properties of modern neural architectures can be partly attributed to the employment of non-linearities (such as ReLU) that exhibit such spectra upon random initialization. Extending the above analysis, we depict that even ReLU networks, when initialized with higher frequencies, fail to converge to minima with good generalization properties.

Finally, we investigate the "flat minima conjecture", an informal hypothesis in the literature that flatness of a minimum is sufficient (Keskar et al., 2016; Ronny Huang et al., 2020; Chaudhari et al., 2019) (but not necessary (Dinh, 2017)) condition for good generalisation. We find that the consistency of the conjecture with experiment is architecture-dependent, while the predictions made using a spectral bias approach are consistent across all examined architectures and problems. Our main contributions are listed below:

- We show that initialization plays a crucial role in governing the implicit regularization of neural networks. Our results advocate for a shift of focus towards initialization in understanding the generalization paradox, which currently revolves primarily around the optimization procedure.
- We conduct experiments in both classification and regression settings. We show that the developed insights are generic across different architectures, nonlinearities, and initialization schemes. Our experiments include practical, deep networks, in contrast to many existing related works.
- We present (empirical) counter-evidence against the flat minima conjecture and show that 1) SGD is not always biased towards flat minima and 2) flat minima do not always correlate with better generalization.

2. Related Works

Implicit regularization Mathematically characterizing implicit regularization of neural networks is at the heart of understanding deep learning. This intriguing phenomenon received increasing attention from the machine learning community after the seminal work by (Zhang et al., 2016), in which they showed that deep models, despite having the capacity to fit even *random* data, demonstrate remarkable generalization properties. Since then, an extensive body of works have tried to characterize implicit regularization through various lenses including training dynamics (Advani et al., 2020; Gidel et al., 2019; Lampinen & Ganguli, 2018; Goldt et al., 2019; Arora et al., 2019), flat minima conjecture (Keskar et al., 2016; Jastrzębski et al., 2017; Wu et al., 2018; Simsekli et al., 2019; Mulayoff & Michaeli, 2020), statistical properties of data (Brutzkus & Globerson, 2020), architectural aspects such as skip connections (He et al., 2020; Huang et al., 2020), and matrix factorization (Gunasekar et al., 2017; Arora et al., 2019; Razin et al., 2021). At a high-level, these works show that deep models implicitly minimize a form of weight norms, regularize derivatives

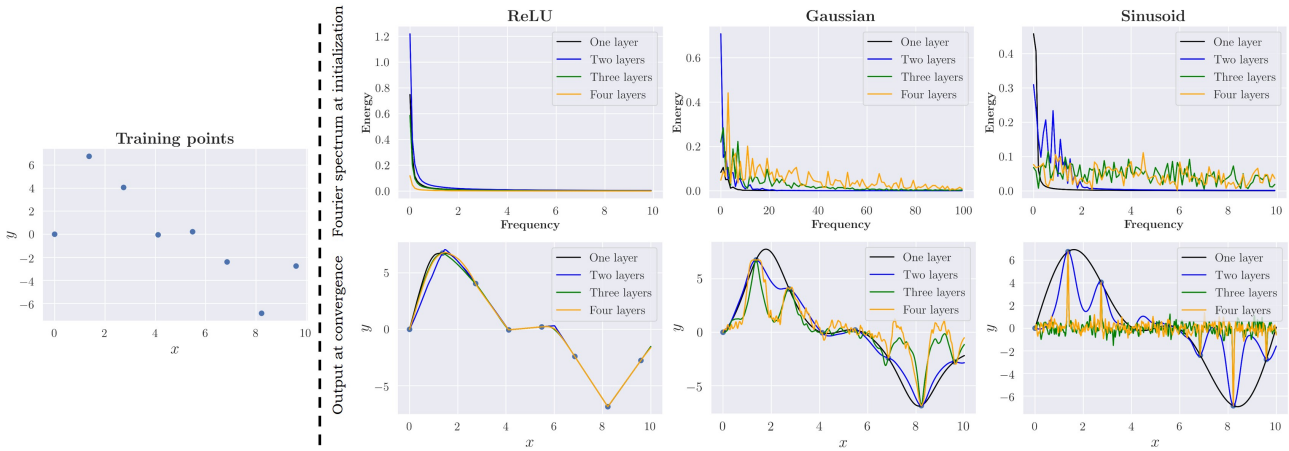

Figure 1. **Implicit neural networks are not implicitly regularized.** The ReLU network keeps converging to smooth solutions despite the increasing depth. In contrast, Gaussian and sinusoidal networks converge to increasingly erratic solutions as the depth is increased. Interestingly, note that the Gaussian and sinusoidal networks add higher frequencies to the spectrum at initialization as the depth is increased, in contrast to the ReLU network.

of the outputs, or analogously, maximize a notion of margin between output classes. However, the center of attention of (almost all) these works is the bias induced by the optimization (SGD) methods. In contrast, we show that the bias of SGD can itself be a source of poor generalization if initialization is not accounted for. Notably, (Zhang et al., 2020b) recently showed that in the NTK regime, for any loss in a general class of functions, the neural networks finds the same global minima—the one that is nearest to the initial value in the parameter space. This result is a strong indication that the generalization of neural networks indeed depends on initialization. Similarly, (Min et al., 2021) recently discussed the role of initialization in the convergence and implicit bias of neural networks. They showed that the rate of convergence of a neural network depends on the level of imbalance of the initialization. Their setting, however, only considered single-hidden-layer linear networks under the square loss. Furthermore, (Zhang et al., 2020a) provide an analysis of the impact of initialization on generalization in the infinite-width chain network setting, and offer a method of initializing at zero to minimise generalization error. In contrast, our analysis is more general, and applies to commonly used, practical networks.

Spectral bias Neural networks tend to learn low frequencies faster. To the best of our knowledge, this peculiar behavior was first systematically demonstrated on ReLU networks by (Xu et al., 2019b) and (Rahaman et al., 2019) in independent studies, and a subsequent theoretical work showed that shallow networks with Tanh activations (Xu, 2018) also admit the same bias. Several recent works have also attempted to characterize the spectral bias of neural networks in different training phases and under various (relatively restrictive) architectural assumptions (Luo et al., 2019; Zhang et al., 2019; Luo et al., 2020). Perhaps, the insights developed by (Zhang et al., 2019) and (Luo et al., 2020) are more closely aligned with some of the conclusions of our work, in which they showed that shallow ReLU networks with infinite width converge to solutions by minimally changing the initial Fourier spectrum. The spectral bias that we prove is slightly weaker precisely due to its generality: namely that sufficiently high frequencies train arbitrarily slowly. However our result applies more generally and its qualitative predictions are borne out by our experiments.

Implicit neural networks Implicit neural networks are a class of fully connected networks that were recently popularized by the seminal work of (Mildenhall et al., 2020). Implicit neural networks either use non-traditional activation functions (Gaussian (Ramasinghe & Lucey, 2021) or Sinusoid (Sitzmann et al., 2020)) or positional embedding layers (Tancik et al., 2020; Zheng et al., 2021). The key difference between implicit neural networks and conventional fully connected networks is that the former can learn high frequency functions more effectively and, thus, can encode natural signals with higher fidelity. Owing to this unique ability, implicit neural networks have penetrated many tasks in computer vision such as texture generation (Henzler et al., 2020; Oechsle et al., 2019; Henzler et al., 2020; Xiang et al., 2021), shape representation (Chen & Zhang, 2019; Deng et al., 2020; Tiwari et al., 2021; Genova et al., 2020; Basher et al., 2021; Mu et al., 2021; Park et al., 2019), and novel view synthesis (Mildenhall et al., 2020; Niemeyer et al., 2020; Saito et al., 2019; Sitzmann et al., 2019; Yu et al., 2021; Pumarola et al., 2021; Rebain et al.,

2021; Martin-Brualla et al., 2021; Wang et al., 2021; Park et al., 2021).

3. Generalization and Fourier spectrum of neural networks

Generalization of neural networks Consider a set of training data $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$ sampled from a distribution $\mathbb{D}$. Given a new set $\{\bar{\mathbf{x}}, \bar{\mathbf{y}}\} \sim \mathbb{D}$, where a neural network f only observes $\{\bar{\mathbf{x}}\}$, the goal is to learn a function such that $f(\bar{\mathbf{x}}) \approx \bar{\mathbf{y}}$. Since $\mathbb{D}$ is unknown, the network tries to learn a function that minimizes an expected cost $\mathbb{E}[\mathcal{L}(f(\mathbf{x}_i), \mathbf{y}_i)]$ over the training data, where $\mathcal{L}$ is a suitable loss function. After training, if the network acts as a good estimator $f : \bar{\mathbf{x}} \to \bar{\mathbf{y}}$, we say that f generalizes well. In classification, usually, a variant of the cross-entropy loss is chosen as $\mathcal{L}$, and in regression, ℓ_1 or ℓ_2 loss is chosen. It should be noted that generalization is entirely a function of $\mathbb{D}$ and thus, cannot be measured without priors on $\mathbb{D}$. In image classification, for instance, a held-out set of validation/testing data is used as a prior on $\mathbb{D}$ to measure the generalization performance. In regression, due to the infinite sampling precision of both input and output spaces, the use of such held-out data becomes less meaningful. Thus, a more practical method of measuring the generalization in a regression setting, at least in an engineering sense, is to measure the "smoothness" of interpolation between training data. That is, we say that a network generalizes well if its output is smooth while fitting the training data (Appendix A.2).

Smooth interpolations and the Fourier spectrum In machine learning and statistics, a "smooth" signal is typically considered a signal with bounded higher-order derivatives. This interpretation stems from the fact that, in practice, noise causes large derivatives and, thus, suppressing higher-order derivatives is equivalent to suppressing noise in a signal, leading to better generalization. A widely used approach to obtain a smooth output signal is regularizing the second-order derivatives. For instance, in spline regression, a weighted sum of second-order derivatives and the square loss is minimized to achieve better generalization (Reinsch, 1967; Craven & Wahba, 1978; Kimeldorf & Wahba, 1970). Interestingly, (Heiss et al., 2019), showed that shallow ReLU networks, when initialized randomly, implicitly regularize the second-order derivatives of the output over a broad class of loss functional, leading to better generalization. Next we show that minimizing the second-order derivatives of a signal is equivalent to minimizing the power of higher frequencies of that particular signal. Consider an absolutely integrable function $g(x)$ and its Fourier transform $\hat{g}(x)$. Then,

$$g(x) = \int_{-\infty}^{\infty} \hat{g}(k) e^{2\pi j k x} dk \tag{1}$$

$$\begin{aligned} \left| \frac{d^2 g(x)}{dx^2} \right| &= |4\pi^2 \int_{-\infty}^{\infty} k^2 \hat{g}(k) e^{2\pi j k x} dk| \\ &\leq |4\pi^2| \int_{-\infty}^{\infty} |k^2 \hat{g}(k)| dk \end{aligned} \tag{2}$$

Therefore, suppressing the higher frequencies of the Fourier spectrum $\hat{g}(k)$ of a signal reduces the upperbound on the magnitude of the second-order derivatives of that particular signal.

Fourier spectrum of a neural network To any integrable function $f : \mathbb{R}^d \to \mathbb{R}$ is associated its *Fourier transform*, given by the formula $\mathcal{F}[f](\mathbf{k}) := \int e^{-i\mathbf{k}\cdot\mathbf{x}} f(\mathbf{x})\, d\mathbf{x}$ (Grafakos, 2008). In particular, a scalar-valued neural network defines a function $f_\theta : \mathbb{R}^d \to \mathbb{R}$, whose Fourier transform makes sense provided f_θ is integrable. We will mollify (set to zero outside of some set) f_θ to take into account data locality, which guarantees integrability. The Fourier transform of a vector-valued network is defined by taking the Fourier transform of each of its component functions.

4. Implicit neural networks do not always generalize well

In this section, we compare implicit neural networks against conventional ReLU networks in regression, and show that the former do not always generalize well. The experiments are described in detail below.

Experiment 1: We utilize fully-connected networks with three types of activation functions: 1) ReLU, 2) Gaussian, and 3) Sinusoidal. We sample 8 sparse points from the signal $3\sin(0.4\pi x) + 5\sin(0.2\pi x)$ and regress them using networks across varying depths. As depicted in Fig. 1, when more capacity is added to the ReLU network via hidden layers, the network keeps converging to a smooth solution as expected. In contrast, Gaussian and Sinusoidal networks showcase worsening interpolations, contradicting the mainstream expectations of implicit regularization. Interestingly, it can be observed that sinusoidal and Gaussian networks add more energy to higher frequencies at initialization as more layers are added. In contrast, ReLU networks tend to have a highly biased spectrum towards lower frequencies irrespective of the depth. All the networks are randomly initialized using Xavier initialization (Glorot & Bengio, 2010). We use SGD to optimize the networks with a learning rate of 1×10^{-4}. The networks consist of 256 neurons in each hidden layer.

Experiment 1 concludes that even with SGD as the optimization algorithm, not all types of networks are implicitly regularized. Instead, the results hint that the initial Fourier

spectrum impacts the generalization performance of a neural network, and the network architecture (activation) plays a crucial role in determining the spectrum. In the upcoming sections, we dig deeper into these insights.

5. The universality of weak spectral bias

Sec. 4 showed that networks with higher frequencies at initialization tend to exhibit poor generalization. However, it is worth investigating if there is indeed a causal link between the two. Intuitively, spectral bias allows us to speculate such a link. That is, one can speculate that the non-smooth interpolations are a result of unwanted residual frequencies after the convergence of lower frequencies. Continuing this line of thought, we present a general proof of a weaker version of spectral bias, which we show to be a universal phenomenon that exists in any parameterized function (which includes the class of all neural networks), given that they are trained with gradient-based optimization methods.

Let $f : \mathbb{R}^p \times \mathbb{R}^d \to \mathbb{R}$ be a parameterized family $\theta \mapsto f_\theta$ of continuous functions $\mathbb{R}^d \to \mathbb{R}$. We assume that the map $(\theta, x) \mapsto f_\theta(x)$ is differentiable almost everywhere, and that the restriction of the (almost everywhere-defined) map $x \mapsto D_\theta f_\theta(x)$ is bounded over any compact set. This setting includes all presently used neural network architectures, with activation functions constrained only to be differentiable almost everywhere.

We care only about the behaviour of f_θ in a neighbourhood of the data. We invoke the *compact data manifold hypothesis*: that the entire data manifold is contained in some compact neighbourhood[1] K. Let g_θ be the extension by zero of f_θ outside of K (our result also holds if one mollifies by a smooth bump function, as in (Luo et al., 2019)) i.e.

$$g_\theta(x) := \begin{cases} f_\theta(x) & \text{if } x \in K, \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Thus g_θ has compact support[2] K and is continuous on K since f_θ is continuous globally. It follows that g_θ is in $L^1(\mathbb{R}^d)$. It follows from the Riemann-Lebesgue lemma (Grafakos, 2008) that the Fourier transform $\mathcal{F}[g_\theta]$ of g_θ vanishes at infinity. The next theorem shows that the same is true of the change $\frac{d}{dt}[g_{\theta(t)}]$ during training, hence that high enough frequencies will be essentially unaffected during training.

Theorem 5.1 (The (weak) spectral bias of differentiably parameterized models)**.** *Let $c : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ be any differentiable cost function, and let $\{x_i\}_{i=1}^n$ be a training set drawn from the data manifold, with corresponding target values $\{y_i\}_{i=1}^n$. Assume that the parameterized function $\theta \mapsto g_\theta$ is trained according to almost-everywhere-defined gradient flow:*

$$\frac{d}{dt} g_{\theta(t)}(x) = -\frac{1}{n} \sum_{i=1}^n \mathcal{K}(\theta(t), x, x_i) \nabla c(g_{\theta(t)}(x_i), y_i), \tag{4}$$

where

$$\mathcal{K}(\theta, x, x') := D_\theta g_\theta(x)\, D_\theta g_\theta(x')^T \tag{5}$$

is the tangent kernel (Jacot et al., 2018b) defined by g_θ. Then the Fourier transform $\mathcal{F}[g_{\theta(t)}]$ evolves according to the differential equation

$$\frac{d}{dt} \mathcal{F}[g_{\theta(t)}](\xi) = -\frac{1}{n} \sum_{i=1}^n \int_{x \in \mathbb{R}^d} e^{-ix\cdot\xi} \mathcal{K}(\theta(t), x, x_i) \nabla c(g_{\theta(t)}(x_i), y_i)\, dx. \tag{6}$$

Moreover, $\frac{d}{dt}\mathcal{F}[g_{\theta(t)}]$ vanishes at infinity: for each $\epsilon > 0$, there exists $\kappa > 0$ such that $\|\xi\| > \kappa$ implies $\left|\frac{d}{dt}\mathcal{F}[g_{\theta(t)}](\xi)\right| < \epsilon$.

A Taylor expansion argument can be used to argue for the same result for discrete-time gradient descent (see Appendix A1). Two remarks are in order regarding Theorem 5.1.

Remark 5.2. For a given architecture, it is desirable to have quantitative bounds on the frequency above which training can be guaranteed to be negligible. Such bounds exist in the literature (Zhang et al., 2019; Luo et al., 2019; 2020), but these bounds make strong architectural assumptions such as limited depth, infinite width or purely chain MLP architectures. While our theorem is *quantitatively* limited, it is *qualitatively* powerful in that it predicts that for *any* learning problem using gradient flow on a parameterized model, sufficiently high frequencies present at initialization will tend to remain after training. Our experiments suggest that for practical neural networks in particular, "sufficiently high" frequencies are far from out of reach and can cause poor generalization.

Remark 5.3. Our use of the tangent kernel in characterising the dynamics of gradient flow are inspired by the seminal work of (Jacot et al., 2018b), which is well-known for hypothesising infinite width for several of its results. In fact, the tangent kernel governs gradient flow dynamics *independently of any architectural assumptions* (beyond the stated differentiability assumption), and in particular, Theorem 5.1 does not require an assumption of infinite width in order to use the tangent kernel. The infinite width hypothesis is invoked in (Jacot et al., 2018b) specifically to give a simple proof of evolution towards a global minimum. We do not attempt any such proof and so do not require the infinite width hypothesis.

[1] A *compact neighbourhood* is a compact set containing a nonempty open set.

[2] The *support* of a function is the smallest closed set containing the set on which the function is nonzero

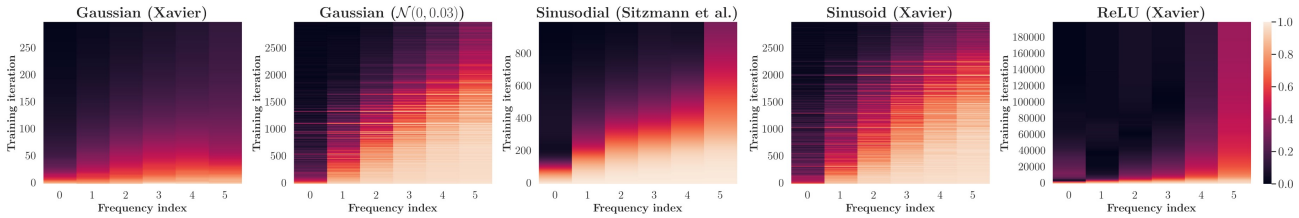

Figure 2. **Spectral bias applies to different network types and initialization schemes.** We measure the convergence of each frequency index as the training progresses. The colors indicate the difference between the ground truth and the predicted frequencies at each index. Xavier and Sitzmann are the initialization schemes proposed by (Glorot & Bengio, 2010) and (Sitzmann et al., 2020), respectively. Note that the convergence-decay rates of frequencies varies across network types and initialization schemes.

Experiement 2: The goal of this experiment is to (partially) empirically validate the above theoretical conclusions. To this end, we use 4-layer deep ReLU, Gaussian, and sinusoid networks where each layer contains 256 neurons. We train the networks on densely sampled points from $g(x) = \sum_{n=1}^{6} \sin(10\pi n x)$. While training, we visualize the convergence of frequency indices of all the networks (Fig. 2). As Theorem 5.1 predicted, all three types of networks exhibit spectral bias. Note that the convergence-decay rates differ across network-types and initialization schemes, which also has an impact on generalization (see Appendix).

In the next section, we show that the initialization plays a key role in generalization and the widely-observed good generalization properties of ReLU networks are merely a consequence of them having biased initial spectra (towards lower frequencies), upon random initialization

6. ReLU networks do not always generalize well

In this section, we show that the initial Fourier spectrum plays a decisive role in governing the implicit regularization of a neural network. Notably, we show that even ReLU networks (which are commonly expected to generalize well) do not always converge to smooth solutions despite training with SGD. We use 4-layer networks where each layer's width is 256 neurons

Experiment 3a: We investigate and analyze the effect of the initial Fourier spectrum on generalization. First, we sample a signal $\sin(\pi x)$ with a step size of 1. Thus, the lowest frequency signal that can fit this set of training points is $\sin(\pi x)$ (Nyquist-Shannon sampling theorem). Then, we randomly initialize a ReLU network using Xavier initialization, so that its initial Fourier spectrum does not contain significant energies above the frequency index $k = 0.5$ (which corresponds to the lowest frequency solution). After training the network over the training points, the network converges to the lowest frequency solution, *i.e.*, $\sin(\pi x)$.

Experiment 3b: We utilize the same training points used in Experiment 4a. However, in this instance, we pre-train the ReLU network on a signal $\sin(10\pi x)$. Note that at this instance, the Fourier spectrum of the network has a spike at $k = 5$, which is above $k = 0.5$. Then, starting from these pre-trained weights, we train the network on the training points.

Experiment 3c: We initialize a Gaussian network with Xavier initialization, so that it contains frequencies above $k = 0.5$. Then, starting from these weights, we train the model on the above training points.

Experiment 3d: We initialize a Gaussian network with a random weight distribution $\mathcal{N}(0, 0.03)$ such that it does not contain frequencies above $k = 0.5$. Then, starting from these weights, we train the model on the training points used in the above experiments.

Fig. 3 visualizes the results. As illustrated, when the spectrum of the ReLU network does not contain frequencies higher than $k = 0.5$, the final spectrum of the network matches with the lowest frequency solution. In contrast, when the initial spectrum of the ReLU network contains frequencies higher than $k = 0.5$, the network adds a spike at $k = 0.5$, but leaves the high-frequency spike untouched as the network has already reached zero train error. This results in a non-smooth (poorly generalized) solution. Interestingly, observe that Gaussian networks also can generalize well if the initial spectrum does not contain higher frequencies. It is vital to note that, however, the convergence-decay rates of frequencies also play an important role. For instance, if the convergence-decay rate is low, higher frequencies begin to get affected *before* the lower frequencies are converged, which can lead to non-smooth solutions (see Appendix). In the next section, we investigate the effect of having high bandwidth spectra at initialization in classification, using popular deep networks.

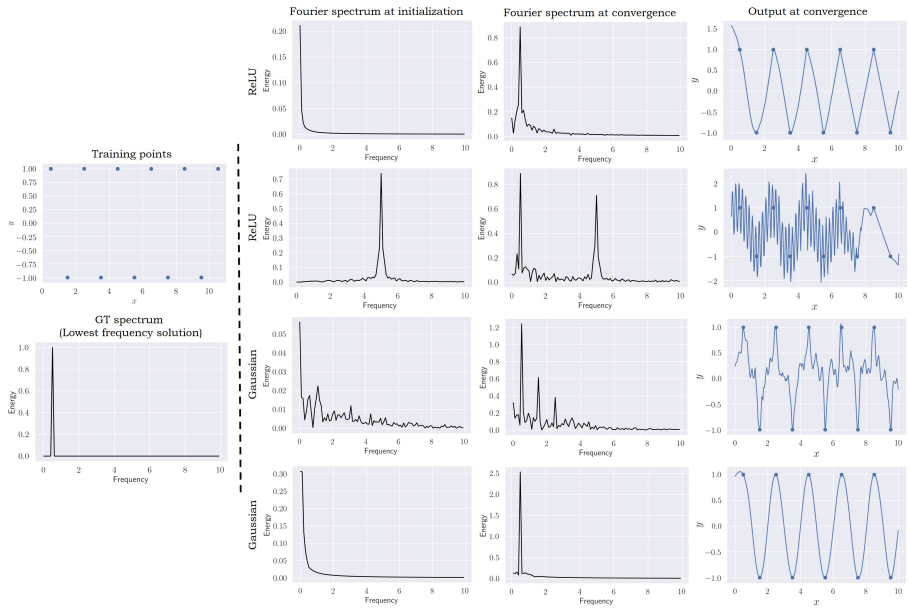

Figure 3. Left block shows sparsely sampled training points from $\sin(\pi x)$ and the corresponding lowest frequency solution that fits the training data. Right block compares generalization corresponding to different networks and initializations. *Top row:* The ReLU network tries to converge to a solution by changing low frequencies at a faster rate due to spectral bias. Consequently, when initialized with no high frequencies, the network ends up converging to the lowest frequency (hence smooth) solution for the training points. *Second row:* **ReLU networks do not always generalize well.** If higher frequencies (than the lowest frequency solution) exist at initialization, ReLU networks reach a solution manipulating only the lower frequencies, resulting in bad interpolations. *Third row:* Same behavior is demonstrated with a Gaussian network. *Fourth row:* **Gaussian networks can generalize well if initialized properly.** Since the network does not contain high frequencies at initialization, it is possible for the network to converge to the lowest frequency solution.

	CIFAR10		CIFAR100		Tiny ImageNet	
Model	Random Init	High B/W init	Random Init	High B/W init	Random Init	High B/W init
VGG11 (Simonyan & Zisserman, 2014)	$84.33 \pm 0.49\%$	$71.94 \pm 0.71\%$	$54.03 \pm 0.71\%$	$41.88 \pm 0.94\%$	$38.87 \pm 0.41\%$	$27.99 \pm 0.73\%$
VGG16 (Simonyan & Zisserman, 2014)	$88.24 \pm 0.12\%$	$71.55 \pm 0.79\%$	$56.86 \pm 0.68\%$	$36.27 \pm 1.92\%$	$40.95 \pm 0.61\%$	$21.77 \pm 0.85\%$
AlexNet (Krizhevsky, 2014)	$80.11 \pm 1.13\%$	$51.31 \pm 0.61\%$	$53.12 \pm 1.01\%$	$41.44 \pm 1.21\%$	$35.56 \pm 0.55\%$	$21.94 \pm 1.66\%$
EfficientNet (Tan & Le, 2019)	$76.78 \pm 0.57\%$	$61.38 \pm 0.46\%$	$43.15 \pm 0.58\%$	$26.83 \pm 0.89\%$	$33.39 \pm 0.42\%$	$18.19 \pm 0.92\%$
DenseNet (Huang et al., 2017)	$86.69 \pm 0.02\%$	$80.86 \pm 0.01\%$	$57.76 \pm 0.24\%$	$46.56 \pm 0.41\%$	$48.86 \pm 0.35\%$	$28.23 \pm 0.27\%$
ResNet-18 (He et al., 2016)	$82.44 \pm 0.15\%$	$68.99 \pm 0.62\%$	$52.14 \pm 0.61\%$	$41.98 \pm 0.59\%$	$43.58 \pm 1.21\%$	$26.73 \pm 0.63\%$
ResNet-50 (He et al., 2016)	$87.18 \pm 0.21\%$	$62.72 \pm 0.47\%$	$54.42 \pm 0.78\%$	$30.56 \pm 0.58\%$	$43.33 \pm 1.43\%$	$28.08 \pm 0.61\%$
SENet (Hu et al., 2018)	$86.31 \pm 0.30\%$	$71.20 \pm 0.35\%$	$58.64 \pm 0.25\%$	$51.56 \pm 0.77\%$	$28.27 \pm 1.33\%$	$24.03 \pm 0.29\%$
ConvMixer (Trockman & Kolter, 2022)	$86.72 \pm 0.97\%$	$49.33 \pm 0.78\%$	$61.20 \pm 0.24\%$	$26.35 \pm 0.99\%$	$45.38 \pm 0.88\%$	$27.77 \pm 0.28\%$

Table 1. **Generalization of deep networks in classification (accuracy $\pm$ std.).** All values reported here are test accuracies where the same model achieved 100% accuracy during training. When the models are initialized with higher bandwidths (pre-trained on random labels), the test accuracy drops. This pattern is consistent across various architectures and datasets. We do not use data augmentation in these experiments and each model is run for five times in each setting.

7. Generalization of deep networks in classification

Sec. 5 affirmed that the spectral bias holds for *any* parameterized model trained using gradient descent. Thus, it is intriguing to explore whether the practical insights we developed thus far extrapolate to popular deep networks that are ubiquitously used. However, for deep networks with high-dimensional inputs (*e.g.*, images), high-bandwidth initialization becomes less straightforward. For instance, consider a network that consumes high dimensional inputs $\mathbf{x} = (x_1, x_2, \ldots, x_n)$. Then, one can hope to directly extend the one dimensional technique we used and train the network on the supervisory signal $\sin(wx_1) \times \sin(wx_2) \times \cdots \times \sin(wx_n)$. However, it is easy to show that in this case, as the dimension of the input grows, the target signal converges to zero.

Let us instead state and justify the following two hypotheses. **First**, we hypothesize that pretraining on random labels will suffice to introduce high frequencies into the resulting function due to the high frequency nature of random noise:

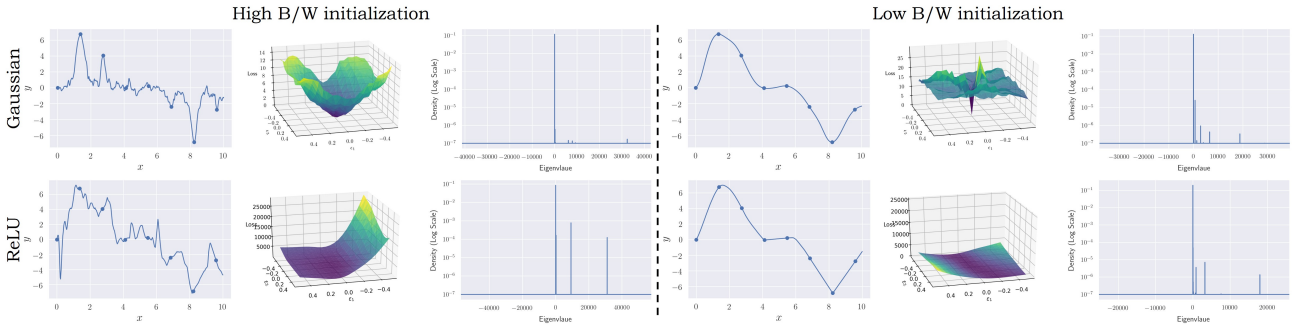

Figure 4. **Flat minima conjecture does not always hold.** The left block and the right block correspond to high bandwidth and low bandwidth initializations, respectively. In each block, from the left column, the interpolations, loss landscapes, and the eigenvalue distribution of the loss-Hessian are illustrated. The loss landscapes are plotted along with the directions of the two largest eigenvalues. As depicted, while our results for the ReLU network are consistent with the conjecture, the Gaussian network behaves in the opposite manner. For more detailed quantitative results, see Table 5.

we empirically justify this using low-dimensional proxy experiments (Appendix A.2) . **Second**, we hypothesize that a high frequency function will generalize poorly in image classification: we believe this to be justified by the manifold hypothesis, which asserts that natural images tend to cluster along smooth manifolds. If these two hypotheses are true, then pretraining a network on random labels before training on real labels will cause worse test performance. This is indeed what we observe as shown next.

Experiment 4 We use 9 popular models for this experiment: VGG16, VGG11, AlexNet, EfficientNet, DenseNet, ResNet-50, ResNet-18, SENet, and ConvMixer. In the first setting, we initialize the models with random weights, train them on the train splits of the datasets, and measure the test accuracy on the test splits. In the next setting, we first pre-train the models on the train split with randomly shuffled labels. Then, starting from the pre-trained weights, we train the models on the train splits of datasets with correct labels and compute the test accuracy on the test splits. The results are depicted in Table 1. Recall that the pre-trained models on random labels yield higher initial bandwidths compared to randomly initialized models. As evident from the results, starting from a higher bandwidth hinders good generalization, validating our previous conclusions. The test accuracies of some models under random weight initialization (Table 1) are slightly lower than the benchmark results reported in the literature. This is because, following (Zhang et al., 2016), we treat data augmentation as an explicit regularization technique and do not use it. In contrast to (Zhang et al., 2016), we consider dropout and batch normalization as architectural aspects and keep them. Nevertheless, it is important to note that in the above experiments, we cannot guarantee that no other adversarial effects will be introduced to the networks other than higher frequencies. We leave precise investigation into this matter to future works.

Model	Random Init	High B/W init	Swapped Init
ResNet-18	82.44 ± 0.15	68.99 ± 0.62	82.91 ± 0.03
ConvMixer	86.72 ± 0.97	49.33 ± 0.78	85.27 ± 0.13

Table 2. The models are able to achieve on-par results compared to the random initialisation scenario when pretrained on swapped labels. This is an indication that the performance drop of the models are a result of high-bandwidth initialization.

Nevertheless, it is intriguing to investigate whether the drop in performance is a result of the high-bandwidth initialization, or if the networks are simply struggling to learn swapped labels. To verify this, we perform an experiment. First, we swap labels randomly and train the model keeping the swapped labels fixed. Then, starting from the above pretrained model, we retrain the model on the correct labels. The results are depicted in Table 2. As shown, the models are able to achieve on-par results compared to the random initialisation scenario.

8. A case against the flat minima conjecture

The “flat minima conjecture" refers to an informal hypothesis present in the literature that convergence of neural network training to a flat minimum is sufficient (Keskar et al., 2016; Chaudhari et al., 2019; Ronny Huang et al., 2020) (but not necessary (Dinh, 2017)) for the network to generalize well. While a good deal of empirical evidence exists to support this conjecture for ReLU networks (see especially (Chaudhari et al., 2019)), we show that the conjecture is not true for Gaussian-activated networks

Experiment 5 We sample four random variables $w_1, w_2 \sim U(0.01, 1), a_1, a_2 \sim U(1, 10)$ and define 20 signals using the sampled variables as $a_1 \sin(2\pi w_1 x) + a_2 \sin(2\pi w_2 x)$. Then, we sample 8 equidistant samples between 0 and 10, and use them as the training points to

train models. We use Gaussian and ReLU networks for this experiment in two settings. In the first setting, we initialize the ReLU network using Xavier initialization and the Gaussian networks with $\mathcal{N}(0, 0.03)$. In this setting, both the networks are able to interpolate the points smoothly. In the other setting, we initialize the ReLU network by pre-training it on $\sin(6\pi x)$ and the Gaussian network with Xavier initialization. In this scenario, both networks demonstrate non-smooth interpolations due to initial high bandwidth. At convergence, we compute the hessian of the loss with respect to the parameters and then compute the eigenvalues and the trace of the hessian. The results are shown in Fig. 4 and Table 5 (Appendix). As evident, the behavior of the Gaussian network is not consistent with the flat minima conjecture.

9. Conclusion

We focus on the effect of initialization on the implicit generalization of neural networks. We reveal that the Fourier spectrum of the network at initialization has a significant impact on the generalization gap. Moreover, we offer evidence against the flat minima conjecture and show that the correlation between the flatness of the minima and the generalization can be architecture-dependent. We empirically validate the generality of our insights across diverse, practical settings.

References

Advani, M. S., Saxe, A. M., and Sompolinsky, H. High-dimensional dynamics of generalization error in neural networks. *Neural Networks*, 132:428–446, 2020.

Ali, A., Dobriban, E., and Tibshirani, R. The implicit regularization of stochastic gradient flow for least squares. In *International conference on machine learning*, pp. 233–244. PMLR, 2020.

Arora, S., Cohen, N., Hu, W., and Luo, Y. Implicit regularization in deep matrix factorization. *Advances in Neural Information Processing Systems*, 32, 2019.

Basher, A., Sarmad, M., and Boutellier, J. Lightsal: Lightweight sign agnostic learning for implicit surface representation. *arXiv preprint arXiv:2103.14273*, 2021.

Brutzkus, A. and Globerson, A. On the inductive bias of a cnn for orthogonal patterns distributions. *arXiv e-prints*, pp. arXiv–2002, 2020.

Chaudhari, P., Choromanska, A., Soatto, S., LeCun, Y., Baldassi, C., Borgs, C., Chayes, J., Sagun, L., and Zecchina, R. Entropy-sgd: Biasing gradient descent into wide valleys. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124018, 2019.

Chen, A., Xu, Z., Geiger, A., Yu, J., and Su, H. Tensorf: Tensorial radiance fields. *arXiv preprint arXiv:2203.09517*, 2022.

Chen, Z. and Zhang, H. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5939–5948, 2019.

Chizat, L. and Bach, F. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. In *Conference on Learning Theory*, pp. 1305–1338. PMLR, 2020.

Chizat, L., Oyallon, E., and Bach, F. On lazy training in differentiable programming. *Advances in Neural Information Processing Systems*, 32, 2019.

Craven, P. and Wahba, G. Smoothing noisy data with spline functions. *Numerische mathematik*, 31(4):377–403, 1978.

Deng, B., Lewis, J. P., Jeruzalski, T., Pons-Moll, G., Hinton, G., Norouzi, M., and Tagliasacchi, A. Nasa neural articulated shape approximation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*, pp. 612–628. Springer, 2020.

Dinh, L. Sharp Minima Can Generalize for Deep Nets. *ICML*, 2017.

Fridovich-Keil, S., Yu, A., Tancik, M., Chen, Q., Recht, B., and Kanazawa, A. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5501–5510, 2022.

Genova, K., Cole, F., Sud, A., Sarna, A., and Funkhouser, T. Local deep implicit functions for 3d shape. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4857–4866, 2020.

Gidel, G., Bach, F., and Lacoste-Julien, S. Implicit regularization of discrete gradient dynamics in linear neural networks. *arXiv preprint arXiv:1904.13262*, 2019.

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256. JMLR Workshop and Conference Proceedings, 2010.

Goldt, S., Advani, M., Saxe, A. M., Krzakala, F., and Zdeborová, L. Dynamics of stochastic gradient descent for two-layer neural networks in the teacher-student setup. *Advances in neural information processing systems*, 32, 2019.

Grafakos, L. *Classical Fourier Analysis*. Number 249 in Graduate Texts in Mathematics. Springer, Second Edition edition, 2008.

Gunasekar, S., Woodworth, B. E., Bhojanapalli, S., Neyshabur, B., and Srebro, N. Implicit regularization in matrix factorization. *Advances in Neural Information Processing Systems*, 30, 2017.

Gunasekar, S., Lee, J., Soudry, D., and Srebro, N. Characterizing implicit bias in terms of optimization geometry. In *International Conference on Machine Learning*, pp. 1832–1841. PMLR, 2018a.

Gunasekar, S., Lee, J. D., Soudry, D., and Srebro, N. Implicit bias of gradient descent on linear convolutional networks. *Advances in Neural Information Processing Systems*, 31, 2018b.

He, F., Liu, T., and Tao, D. Why resnet works? residuals generalize. *IEEE transactions on neural networks and learning systems*, 31(12):5349–5362, 2020.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Heiss, J., Teichmann, J., and Wutte, H. How implicit regularization of neural networks affects the learned function–part i. *arXiv*, pp. 1911–02903, 2019.

Henzler, P., Mitra, N. J., and Ritschel, T. Learning a neural 3d texture space from 2d exemplars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8356–8364, 2020.

Hochreiter, S. and Schmidhuber, J. Simplifying neural nets by discovering flat minima. *Advances in neural information processing systems*, 7, 1994.

Hochreiter, S. and Schmidhuber, J. Flat minima. *Neural computation*, 9(1):1–42, 1997.

Hu, J., Shen, L., and Sun, G. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141, 2018.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

Huang, K., Wang, Y., Tao, M., and Zhao, T. Why do deep residual networks generalize better than deep feedforward networks?—a neural tangent kernel perspective. *Advances in neural information processing systems*, 33: 2698–2709, 2020.

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. *arXiv preprint arXiv:1806.07572*, 2018a.

Jacot, A., Gabriel, F., and Hongler, C. Neural Tangent Kernel: Convergence and Generalization in Neural Networks. *NeurIPS*, 2018b.

Jastrzębski, S., Kenton, Z., Arpit, D., Ballas, N., Fischer, A., Bengio, Y., and Storkey, A. Three factors influencing minima in sgd. *arXiv preprint arXiv:1711.04623*, 2017.

Ji, Z. and Telgarsky, M. Gradient descent aligns the layers of deep linear networks. *arXiv preprint arXiv:1810.02032*, 2018.

Ji, Z. and Telgarsky, M. The implicit bias of gradient descent on nonseparable data. In *Conference on Learning Theory*, pp. 1772–1798. PMLR, 2019.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

Kimeldorf, G. S. and Wahba, G. A correspondence between bayesian estimation on stochastic processes and smoothing by splines. *The Annals of Mathematical Statistics*, 41 (2):495–502, 1970.

Kohler, M., Krzyżak, A., and Schäfer, D. Application of structural risk minimization to multivariate smoothing spline regression estimates. *Bernoulli*, pp. 475–489, 2002.

Krizhevsky, A. One weird trick for parallelizing convolutional neural networks. *arXiv preprint arXiv:1404.5997*, 2014.

Lampinen, A. K. and Ganguli, S. An analytic theory of generalization dynamics and transfer learning in deep linear networks. *arXiv preprint arXiv:1809.10374*, 2018.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Li, Y., Ma, T., and Zhang, H. Algorithmic regularization in over-parameterized matrix sensing and neural networks with quadratic activations. In *Conference On Learning Theory*, pp. 2–47. PMLR, 2018.

Luo, T., Ma, Z., Xu, Z.-Q. J., and Zhang, Y. Theory of the frequency principle for general deep neural networks. *arXiv preprint arXiv:1906.09235*, 2019.

Luo, T., Ma, Z., Xu, Z.-Q. J., and Zhang, Y. On the exact computation of linear frequency principle dynamics and its generalization. *arXiv preprint arXiv:2010.08153*, 2020.

Ma, C., Wang, K., Chi, Y., and Chen, Y. Implicit regularization in nonconvex statistical estimation: Gradient descent converges linearly for phase retrieval and matrix completion. In *International Conference on Machine Learning*, pp. 3345–3354. PMLR, 2018.

Martin-Brualla, R., Radwan, N., Sajjadi, M. S., Barron, J. T., Dosovitskiy, A., and Duckworth, D. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7210–7219, 2021.

Mei, S., Misiakiewicz, T., and Montanari, A. Mean-field theory of two-layers neural networks: dimension-free bounds and kernel limit. In *Conference on Learning Theory*, pp. 2388–2464. PMLR, 2019.

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, pp. 405–421. Springer, 2020.

Min, H., Tarmoun, S., Vidal, R., and Mallada, E. On the explicit role of initialization on the convergence and implicit bias of overparametrized linear networks. In *International Conference on Machine Learning*, pp. 7760–7768. PMLR, 2021.

Moroshko, E., Woodworth, B. E., Gunasekar, S., Lee, J. D., Srebro, N., and Soudry, D. Implicit bias in deep linear classification: Initialization scale vs training accuracy. *Advances in neural information processing systems*, 33: 22182–22193, 2020.

Mu, J., Qiu, W., Kortylewski, A., Yuille, A., Vasconcelos, N., and Wang, X. A-sdf: Learning disentangled signed distance functions for articulated shape representation. *arXiv preprint arXiv:2104.07645*, 2021.

Mulayoff, R. and Michaeli, T. Unique properties of flat minima in deep networks. In *International Conference on Machine Learning*, pp. 7108–7118. PMLR, 2020.

Niemeyer, M., Mescheder, L., Oechsle, M., and Geiger, A. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3504–3515, 2020.

Oechsle, M., Mescheder, L., Niemeyer, M., Strauss, T., and Geiger, A. Texture fields: Learning texture representations in function space. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4531–4540, 2019.

Oymak, S. and Soltanolkotabi, M. Overparameterized nonlinear learning: Gradient descent takes the shortest path? In *International Conference on Machine Learning*, pp. 4951–4960. PMLR, 2019.

Park, J. J., Florence, P., Straub, J., Newcombe, R., and Lovegrove, S. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 165–174, 2019.

Park, K., Sinha, U., Barron, J. T., Bouaziz, S., Goldman, D. B., Seitz, S. M., and Martin-Brualla, R. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5865–5874, 2021.

Pumarola, A., Corona, E., Pons-Moll, G., and Moreno-Noguer, F. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10318–10327, 2021.

Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F., Bengio, Y., and Courville, A. On the spectral bias of neural networks. In *International Conference on Machine Learning*, pp. 5301–5310. PMLR, 2019.

Ramasinghe, S. and Lucey, S. Beyond periodicity: Towards a unifying framework for activations in coordinate-mlps. *arXiv preprint arXiv:2111.15135*, 2021.

Rangamani, A., Nguyen, N. H., Kumar, A., Phan, D., Chin, S. H., and Tran, T. D. A scale invariant flatness measure for deep network minima. *arXiv preprint arXiv:1902.02434*, 2019.

Razin, N., Maman, A., and Cohen, N. Implicit regularization in tensor factorization. In *International Conference on Machine Learning*, pp. 8913–8924. PMLR, 2021.

Rebain, D., Jiang, W., Yazdani, S., Li, K., Yi, K. M., and Tagliasacchi, A. Derf: Decomposed radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14153–14161, 2021.

Reinsch, C. H. Smoothing by spline functions. *Numerische mathematik*, 10(3):177–183, 1967.

Ronny Huang, W., Emam, Z., Goldblum, M., Fowl, L., Terry, J. K., Huang, F., and Goldstein, T. Understanding generalization through visualizations. *NeurIPS Workshop*, 2020.

Saito, S., Huang, Z., Natsume, R., Morishima, S., Kanazawa, A., and Li, H. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2304–2314, 2019.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Simsekli, U., Sagun, L., and Gurbuzbalaban, M. A tail-index analysis of stochastic gradient noise in deep neural networks. In *International Conference on Machine Learning*, pp. 5827–5837. PMLR, 2019.

Sitzmann, V., Zollhöfer, M., and Wetzstein, G. Scene representation networks: Continuous 3d-structure-aware neural scene representations. *arXiv preprint arXiv:1906.01618*, 2019.

Sitzmann, V., Martel, J., Bergman, A., Lindell, D., and Wetzstein, G. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33, 2020.

Soudry, D., Hoffer, E., Nacson, M. S., Gunasekar, S., and Srebro, N. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878, 2018.

Tan, M. and Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pp. 6105–6114. PMLR, 2019.

Tancik, M., Srinivasan, P. P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J. T., and Ng, R. Fourier features let networks learn high frequency functions in low dimensional domains. *arXiv preprint arXiv:2006.10739*, 2020.

Tiwari, G., Sarafianos, N., Tung, T., and Pons-Moll, G. Neural-gif: Neural generalized implicit functions for animating people in clothing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11708–11718, 2021.

Trockman, A. and Kolter, J. Z. Patches are all you need? *arXiv preprint arXiv:2201.09792*, 2022.

Tsuzuku, Y., Sato, I., and Sugiyama, M. Normalized flat minima: Exploring scale invariant definition of flat minima for neural networks using pac-bayesian analysis. In *International Conference on Machine Learning*, pp. 9636–9647. PMLR, 2020.

Wang, Z., Wu, S., Xie, W., Chen, M., and Prisacariu, V. A. Nerf–: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021.

Wei, C., Lee, J. D., Liu, Q., and Ma, T. Regularization matters: Generalization and optimization of neural nets vs their induced kernel. *Advances in Neural Information Processing Systems*, 32, 2019.

Wu, L., Ma, C., et al. How sgd selects the global minima in over-parameterized learning: A dynamical stability perspective. *Advances in Neural Information Processing Systems*, 31, 2018.

Wu, X., Dobriban, E., Ren, T., Wu, S., Li, Z., Gunasekar, S., Ward, R., and Liu, Q. Implicit regularization of normalization methods. *arXiv preprint arXiv:1911.07956*, 2019.

Xiang, F., Xu, Z., Hasan, M., Hold-Geoffroy, Y., Sunkavalli, K., and Su, H. Neutex: Neural texture mapping for volumetric neural rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7119–7128, 2021.

Xu, Z. J. Understanding training and generalization in deep learning by fourier analysis. *arXiv preprint arXiv:1808.04295*, 2018.

Xu, Z.-Q. J., Zhang, Y., Luo, T., Xiao, Y., and Ma, Z. Frequency principle: Fourier analysis sheds light on deep neural networks. *arXiv preprint arXiv:1901.06523*, 2019a.

Xu, Z.-Q. J., Zhang, Y., and Xiao, Y. Training behavior of deep neural network in frequency domain. In *International Conference on Neural Information Processing*, pp. 264–274. Springer, 2019b.

Yao, Z., Gholami, A., Keutzer, K., and Mahoney, M. W. Pyhessian: Neural networks through the lens of the hessian. In *2020 IEEE international conference on big data (Big data)*, pp. 581–590. IEEE, 2020.

Yu, A., Ye, V., Tancik, M., and Kanazawa, A. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4578–4587, 2021.

Yun, C., Krishnan, S., and Mobahi, H. A unifying view on implicit bias in training linear neural networks. *arXiv preprint arXiv:2010.02501*, 2020.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. *arXiv e-prints*, pp. arXiv–1611, 2016.

Zhang, Y., Xu, Z.-Q. J., Luo, T., and Ma, Z. Explicitizing an implicit bias of the frequency principle in two-layer neural networks. *arXiv preprint arXiv:1905.10264*, 2019.

Zhang, Y., Xu, Z.-Q. J., Luo, T., and Ma, Z. A type of generalization error induced by initialization in deep neural networks. In *Proceedings on Machine Learning Research*, pp. 144–164, 2020a.

Zhang, Y., Xu, Z.-Q. J., Luo, T., and Ma, Z. A type of generalization error induced by initialization in deep neural networks. In *Mathematical and Scientific Machine Learning*, pp. 144–164. PMLR, 2020b.

Zheng, J., Ramasinghe, S., and Lucey, S. Rethinking positional encoding. *arXiv preprint arXiv:2107.02561*, 2021.

A. Appendix

A.1. Proof of Theorem 5.1

Proof. The evolution in parameter space is described by the differential equation

$$\frac{d}{dt}\theta(t) = -\frac{1}{n}\sum_{i=1}^{n} D_\theta g_{\theta(t)}(x_i)^T \nabla c(g_{\theta(t)}(x_i), y_i).$$

The evolution of the corresponding function $g_{\theta(t)}$ is given by pushing this differential equation forward to function space by acting on both sides with the derivative $D_\theta g_{\theta(t)}(x)$:

$$\begin{aligned}\frac{d}{dt}g_{\theta(t)}(x) = D_\theta g_{\theta(t)}(x)\frac{d}{dt}\theta(t) &= -\frac{1}{n}\sum_{i=1}^{n} D_\theta g_{\theta(t)}(x)\, D_\theta g_{\theta(t)}(x_i)^T \nabla c(g_{\theta(t)}(x_i), y_i) \\ &= -\frac{1}{n}\sum_{i=1}^{n} \mathcal{K}(\theta, x, x_i)\nabla c(g_{\theta(t)}(x_i), y_i),\end{aligned}$$

where $\mathcal{K}$ the extension of the tangent kernel associated to f_θ by zero outside of the compact neighbourhood K of the data manifold, i.e.

$$\mathcal{K}(\theta, x, x') = \begin{cases} D_\theta f_\theta(x)\, D_\theta f_\theta(x')^T, & \text{if } x, x' \in K \\ 0, & \text{otherwise.}\end{cases}$$

The evolution equation for $\mathcal{F}[g_{\theta(t)}]$ follows easily from the Liebniz integral rule:

$$\frac{d}{dt}\mathcal{F}[g_{\theta(t)}] = \mathcal{F}\left[\frac{d}{dt}g_{\theta(t)}\right].$$

Now, by our hypothesis on f_θ that $x \mapsto D_\theta f_\theta(x)$ is bounded over compact sets, one has that $x \mapsto \mathcal{K}(\theta, x, x_i)$ is L^1 for each i, hence that $\frac{d}{dt}g_{\theta(t)}$ is an L^1 function. By the Riemann-Lebesgue lemma its Fourier transform vanishes at infinity as stated. □

The same result can be argued for discrete-time gradient descent as follows. At a given time step T, the gradient update is given by the equation

$$\theta_{T+1} - \theta_T = -\frac{\eta}{n}\sum_{i=1}^{n} D_\theta f_{\theta_T}(x_i)^T \nabla c(f_{\theta_T}(x_i)),$$

where η is the step size. One wishes to show that the difference $x \mapsto f_{\theta_{T+1}}(x) - f_{\theta_T}(x)$, extended by zero for x outside of the compact data manifold K, has Fourier transform vanishing at infinity. To first order in η, one can approximate this difference by

$$-\frac{\eta}{n}\sum_{i=1}^{n} D_\theta f_{\theta_T}(x) D_\theta f_{\theta_T}(x_i)^T \nabla c(f_{\theta_T}(x_i)),$$

again extended by zero for x outside of K. Spectral bias for gradient descent then follows (at least approximately, for $\eta << 1$) from the same Riemann-Lebesgue argument that we used for gradient flow.

A.2. Smoothness, generalization, and the the empirical risk minimization (ERM)

The ERM framework provides a well-established framework for studying the generalization in learnable models. The smoothness is a property which stems from the empirical risk minimization framework, and has been used since the earliest days of ML to quantify generalization (in regression). In summary, given a set of hypotheses (models) that minimizes the empirical risk (with training data), the ERM framework prefers a solution that minimizes the true risk (with respect to the actual data distribution) with a higher probability. When extra prior knowledge is unavailable on the true data distribution, ERM suggests that the best solution would be the one that minimizes the least complex solution that minimizes the empirical risk (under the realizability assumption). This can be primarily achieved using two regularization techniques: 1) regularizing the parameters of the model or 2) regularizing the function output itself. Popular regularizations on NNs, Lasso regression,

Ridge regression etc. fall into the first category, and spline, polynomial regression with regularized derivatives fall into the second category (Reinsch, 1967; Kimeldorf & Wahba, 1970; Craven & Wahba, 1978; Kohler et al., 2002). A more recent example is (Heiss et al., 2019). It should be noted that both these techniques lead to smooth solutions with bounded (higher-order) derivatives.

The intuition for this partially stems from the fact that reducing the bandwidth of a signal can be considered as minimizing noise, whereas noise corresponds to higher frequencies in natural signals. Almost every spectral-bias-related recent work also uses low-frequency solutions, hence solutions with bounded second-order derivatives, as a proxy for measuring generalization (Xu et al., 2019a;b). A few application-specific examples would be recent Neural Radiance Field works (Fridovich-Keil et al., 2022; Chen et al., 2022), where smooth (tri-linear) interpolation is used to generalize to unseen coordinates.

A.3. Initializing deep networks with higher bandwidths

Initializing deep classification networks – that consume high dimensional inputs such as images – such that they have higher bandwidths is not straightforward. Therefore, we explore alternative ways to initialize networks with higher bandwidths in low-dimensional settings, and extrapolate the learned insights to higher dimensions.

For all the experiments, we consider a fully connected 4-layer ReLU network with 1-dimensional inputs. First, we sample a set of values from white Gaussian noise, and train the network with these target values using MSE loss. In the second experiment, we threshold the sampled values to obtain a set of binary labels, and then train the network with binary cross-entropy loss. For the third experiment, we use a network with four outputs. Then, we separate the sampled values into four bins, and obtain four labels. Then, we train the network with cross-entropy loss. We compute the Fourier spectra of each of the trained networks after convergence. The results are shown in Fig.5.

As depicted, we can use mean squared error (MSE) or cross-entropy (CE) loss along with random labels to initialize the networks with higher bandwidth. However, we observed that, in practice, deep networks take an infeasible amount of time to converge with the MSE loss. Therefore, we use cross-entropy loss with random labels to initialize the networks in image classification settings.

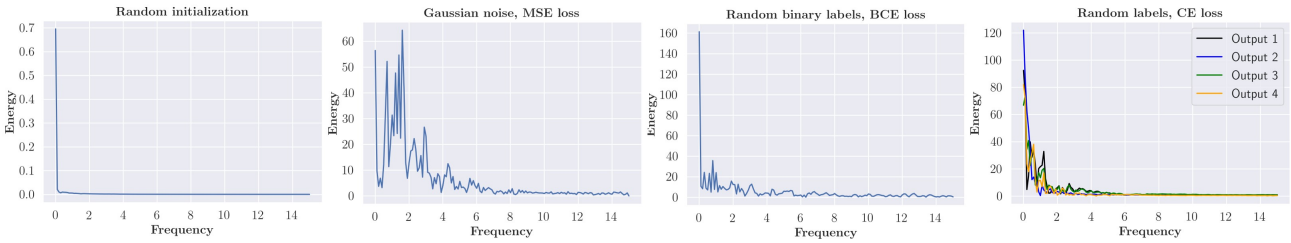

Figure 5. We visualize the spectra of networks after training them with different loss functions and label sampling schemes (the rightmost three plots). All shown methods are able to obtain higher bandwidths than random initialization (leftmost plot). Note that the scale in the $y-$axis is different for each plot. However, in practice, deep classification networks take an infeasible amount of time to converge with MSE loss. Hence, we chose random labels with cross-entropy loss to initialize the deep classification networks with higher bandwidths.

In order to verify that training with random labels indeed induces higher bandwidths on deep classification networks, we visualize the histograms of their first order gradients of the averaged outputs w.r.t. the inputs. It is straightforward to show that (similar to second-order gradients) higher first-order gradients lead to higher bandwidth. For simplicity, consider a function $f : \mathbb{R} \rightarrow \mathbb{R}$. Then,

$$f(x) = \int_{\infty}^{\infty} \hat{f}(k) e^{2\pi i k x} dk$$

It follows that,

$$|\frac{df(x)}{dx}| = |2\pi i \int_{\infty}^{\infty} k\hat{f}(k)e^{2\pi ikx}dk| \tag{7}$$

$$\leq |2\pi| \int_{\infty}^{\infty} |k\hat{f}(k)|dk. \tag{8}$$

Therefore,

$$\max_{x\in\epsilon} |\frac{df(x)}{dx}| \leq |2\pi| \int_{\infty}^{\infty} |k\hat{f}(k)|dk. \tag{9}$$

This conclusion can be directly extrapolated to higher-dimensional inputs, where the Fourier transform is also high dimensional. Hence, we feed a batch of images to the networks, and calculate the gradients of the averaged output layer with respect to the input image pixels. Then, we plot the histograms of the gradients (Fig. 6). As illustrated, training with random labels induces higher gradients, and thus, higher bandwidth. Table. 3 compares generalization of deep networks on ImageNet.

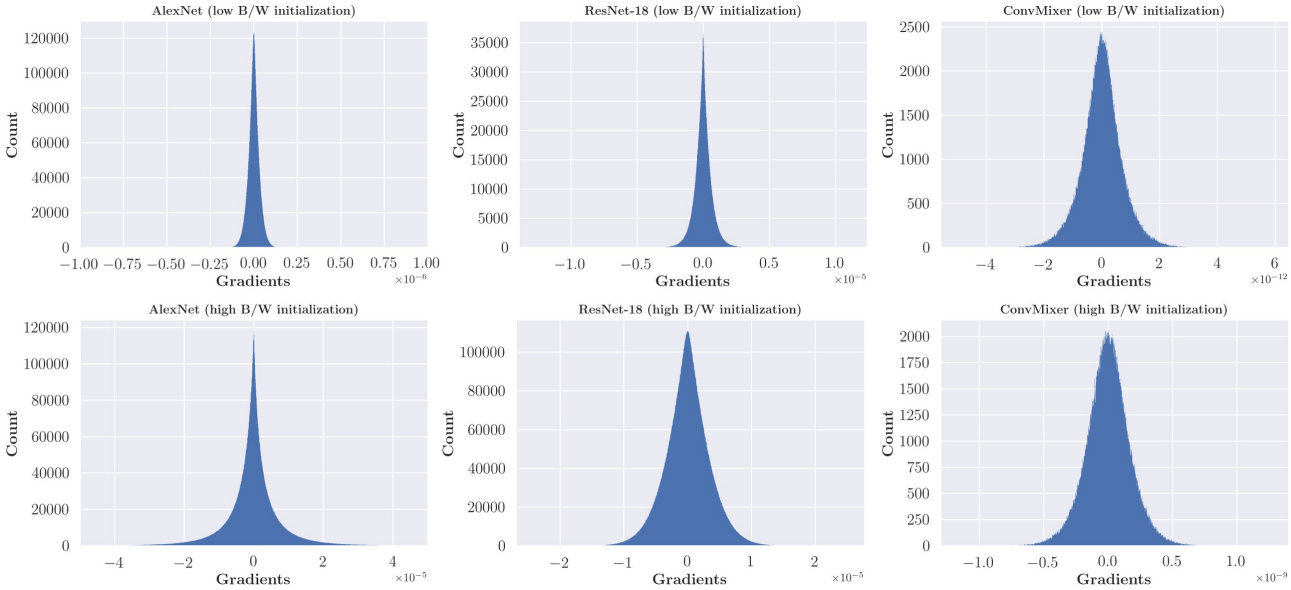

Figure 6. The histograms of the first-order gradients of the outputs with respect to the inputs (a batch of training images) are shown. Low and high bandwidth initializations correspond to Xavier initialization and pre-training with random labels, respectively. Not that the $x-$axis scales are different in each plot. As depicted, training with random labels leads to higher gradients, validating that it indeed leads to higher bandwidths.

ImageNet				
Model	Random initialization		High B/W initialization	
	Train accuracy	Test accuracy	Train accuracy	Test accuracy
VGG16	100%	68.19%	100%	55.48%
ResNet-18	100%	66.93%	100%	49.17%
ConvMixer	100%	74.19%	100%	45.68%

Table 3. **Generalization of deep networks in classification over ImageNet.** When the models are initialized with higher bandwidths (pre-trained on random labels), the test accuracy drops. We do not use data augmentation in these experiments. We only use three models for this experiment due to the extensive resource usage when training on random labels over ImageNet.

A.4. Convergence-decay rates of frequencies matter for generalization

Earlier, we showed that although all neural networks admit spectral bias, the convergence-decay rates of frequencies change across network types and initialization schemes. Below, we show that these decay rates play an essential role in generalization.

We use a Gaussian network for this experiment. We initialize two instances of the network by 1) using a weight distribution $\mathcal{N}(0, 0.03)$, and 2) pre-training the network on a DC signal. In both instances, the network has low bandwidth. Then, we train the network on sparse training data sampled from $3\sin(0.4\pi x) + 5\sin(0.2\pi x)$. The results are shown in Fig. 7. Observe that although both networks start from low bandwidth, they exhibit different generalization properties. This is because, having a lower convergence-decay hinders smooth interpolations even in cases where the networks have low initial bandwidth. This is expected, since then, the optimization will begin to affect the higher frequencies before the lower frequencies are converged.

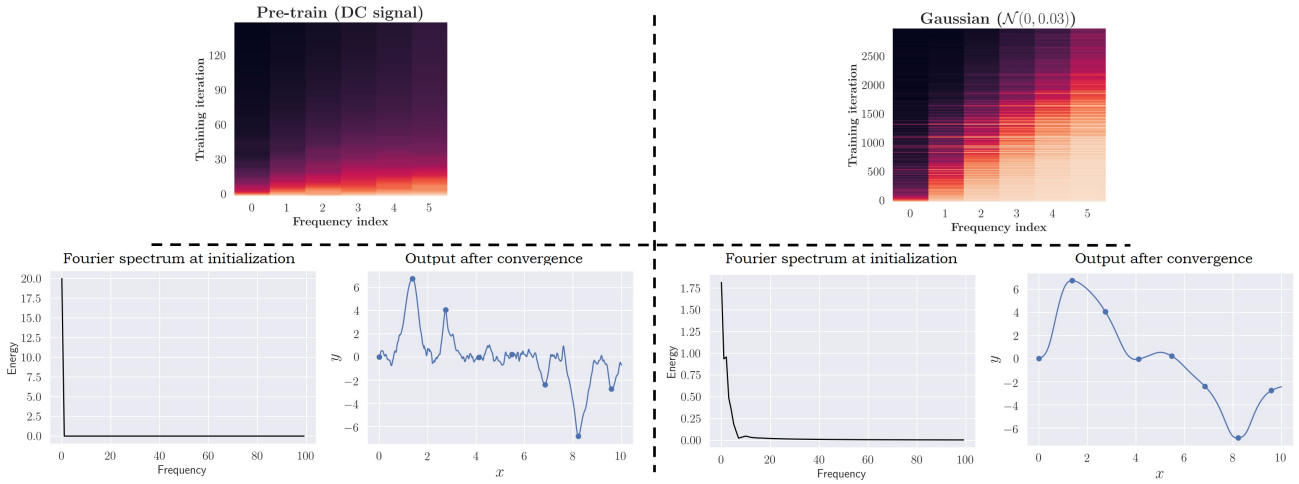

Figure 7. **The effect of convergence-decay rate of frequencies on generalization.** *Left block*: We pre-train a Gaussian network on a DC signal to obtain low initial bandwidth. Nevertheless, the network still converges to a non-smooth solution. *Right block*: The Gaussian network is initialized using a random Gaussian distribution ($\mathcal{N}(0, 0.03)$). This method also leads to lower bandwidth. However, in this scenario, the network is able to converge to a smooth solution. At the top, the convergence of frequency components – starting from the corresponding initialization – is shown when training on a signal $g(x) = \sum_{n=1}^{6} \sin(10\pi n x)$. Note that a lower convergence decay rate leads to bad generalization.

To further verify this, we conduct another experiment; see Fig. 8.

A.5. Analyzing the loss landscapes

The flat minima conjecture has been studied since the early work of (Hochreiter & Schmidhuber, 1994) and (Hochreiter & Schmidhuber, 1997). More recently, empirical works showed that the generalization of deep networks is related to the flatness of the minima it is converged to during training (Chaudhari et al., 2019; Keskar et al., 2016). In order to measure the flatness of loss landscapes, different metrics have been proposed (Tsuzuku et al., 2020; Rangamani et al., 2019; Hochreiter & Schmidhuber, 1994; 1997). In particular, (Chaudhari et al., 2019) and (Keskar et al., 2016) showed that minima with low Hessian spectral norm generalize better. In this paper also, we use Hessian-related metrics to measure the flatness. Since the spectral norm alone is not ideal for analyzing the loss landscape of models with a large number of parameters, we also compute the trace and the expected eigenvalue of the Hessian. For computing the Hessian, we use the library provided by (Yao et al., 2020). Fig 9 and Table 4 depict a comparison of loss landscapes in several deep models. Note that our proposed high B/W initialization scheme provides an ideal platform to compare the loss landscapes with different generalization properties.

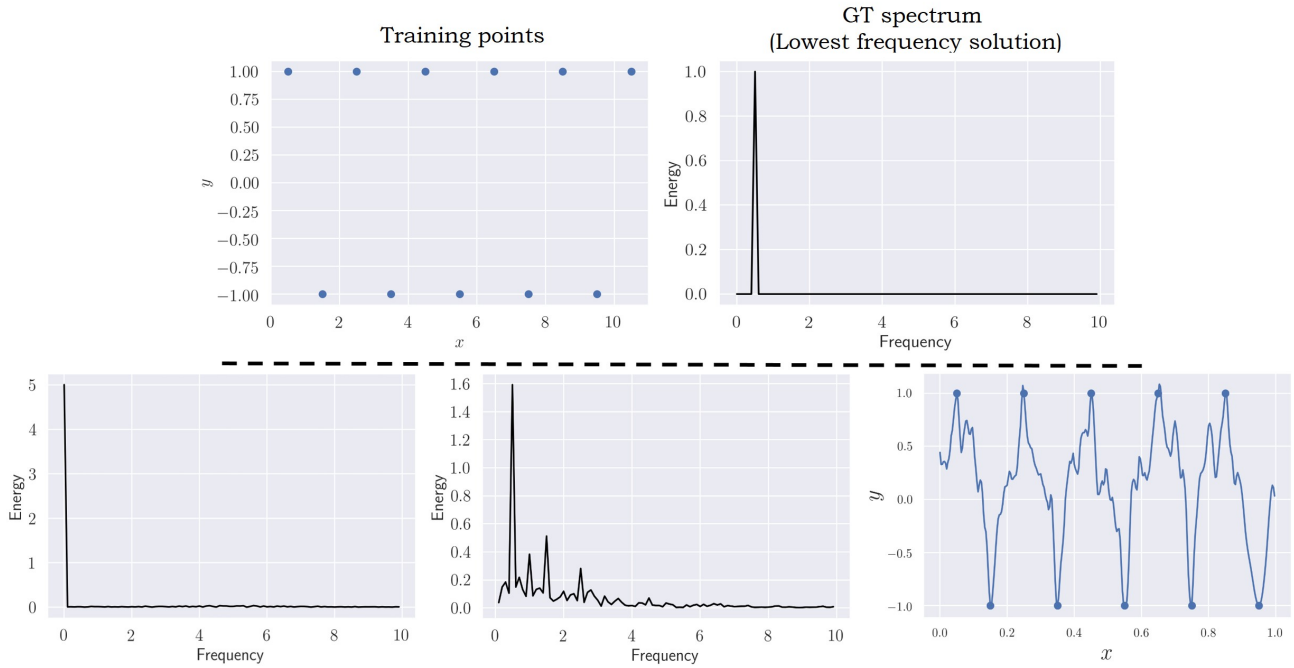

Figure 8. The top block shows sparsely sampled training points from $\sin(\pi x)$ and the corresponding lowest frequency solution that fits the training data. The bottom block shows the spectra of a Gaussian network initialized by pre-training on a DC signal. Even though the network adds a spike at the lowest frequency solution, higher frequencies are also added to the spectrum due to the low convergence-decay rate. This results in a non-smooth interpolation.

Model	Hessian-trace	Spectral norm
ResNet-18 (low B/W)	13560.76	2805.47
ResNet-18 (high B/W)	28614.19	4121.36
VGG-16 (low B/W)	10102.51	1112.07
VGG-16 (high B/W)	14483.90	3214.57
ConvMixer (low B/W)	0.3242	0.028
ConvMixer (high B/W)	3.49	0.445

Table 4. Quantitative comparison of the flatness of minima in deep networks. Note that Note that ReLU networks exhibit behaviour consistent with the flat minima conjecture.

Model	Initialization	Hessian trace	$\mathbb{E}[\epsilon]$	Spectral norm
ReLU	High B/W	134213.36	0.95	257875.23
ReLU	Low B/W	31110.73	0.04	49781.58
Gaussian	High B/W	40478.82	0.21	12596.89
Gaussian	Low B/W	59447.46	0.32	26519.66

Table 5. The trace, expected eigenvalue ($\mathbb{E}[\epsilon)$], and the spectral norm of the loss-Hessian are shown (averaged over 20 signals). Higher values indicate a sharper minimum. As illustrated, while the ReLU network obeys the flat minima conjecture, the Gaussian network behaves oppositely.

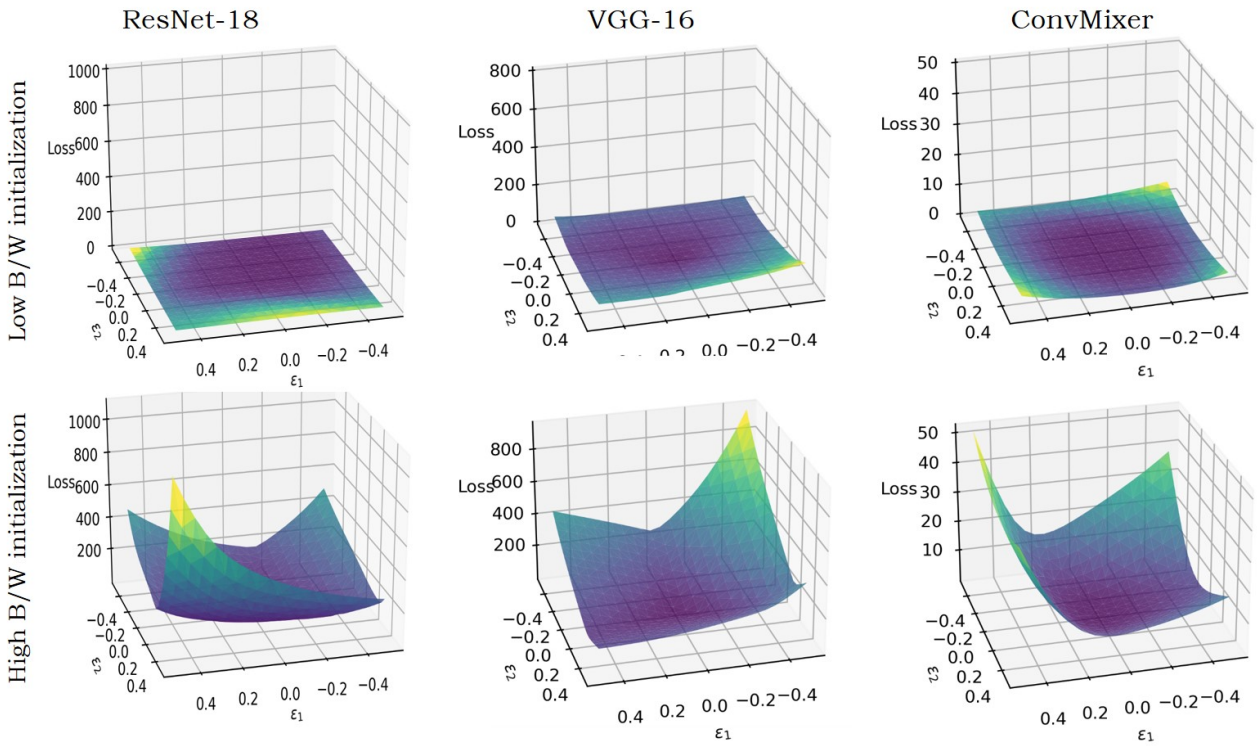

Figure 9. **Loss landscapes of deep networks trained on CIFAR10.** The proposed high B/W initialization scheme provides an ideal platform to compare the flatness of minima with different generalization properties. Note that ReLU networks exhibit behaviour consistent with the flat minima conjecture.