Investigating Generalization by Controlling Normalized Margin

Alexander R. Farhang [1] **Jeremy Bernstein** [1] **Kushal Tirumala** [1] **Yang Liu** [2] **Yisong Yue** [1 2]

Abstract

Weight norm $\|w\|$ and margin γ participate in learning theory via the normalized margin $\gamma/\|w\|$. Since standard neural net optimizers do not control normalized margin, it is hard to test whether this quantity causally relates to generalization. This paper designs a series of experimental studies that explicitly control normalized margin and thereby tackle two central questions. First: *does normalized margin always have a causal effect on generalization?* The paper finds that *no*—networks can be produced where normalized margin has seemingly no relationship with generalization, counter to the theory of Bartlett et al. (2017). Second: *does normalized margin ever have a causal effect on generalization?* The paper finds that *yes*—in a standard training setup, test performance closely tracks normalized margin. The paper suggests a Gaussian process model as a promising explanation for this behavior.

1. Introduction

Despite significant progress, a complete explanation of the remarkable generalization capabilities of neural networks remains an open problem. Experimental studies often seek *complexity measures* (Pérez & Louis, 2020) or optimization and architectural *hyperparameters* (Keskar et al., 2017) with explanatory power. But due to both the number of moving parts in a deep learning system and the cost of experimentation, unpacking the underlying effects is challenging.

One significant hurdle to a full scientific understanding of generalization is the presence of numerous potential *confounders*. Even if a complexity measure, say, is strongly *correlated* with generalization, this does not imply a causal link (Jiang et al., 2020). Inspired by this observation, this paper singles out a quantity that is implicated by many theories as an important factor in generalization—normalized margin—and attempts to pin down its causal link to generalization. This quest is broken up into two separate sub-questions.

First, some theories suggest that normalized margin may have a very broad controlling effect on generalization (Bartlett et al., 2017). To study that idea, this paper asks:

⟨Q1⟩ ***Does normalized margin always have a causal effect on generalization?***

In other words: is a notion of normalized margin *sufficient* to explain generalization? Is it the dominant factor? Or are there *counterexamples*: settings where normalized margin is uninformative?

Second, some theories address the *typical* behavior of function spaces rather than the *worst case* (McAllester, 1999). Perhaps normalized margin has a causal effect in these more typical settings. Consequently, this paper asks:

⟨Q2⟩ ***Does normalized margin ever have a causal effect on generalization?***

In other words: is a notion of normalized margin *necessary* to build a complete picture of generalization? As posed, this question can be answered by finding *positive examples* of settings where normalized margin has a causal effect.

To tackle these questions, this paper designs a series of experimental studies that take care to control both weight norms and margins of the learned predictors, resulting in the ability to target specific normalized margin distributions during training. The studies consider both *spectrally-normalized* and *Frobenius-normalized* margin distributions.

In answer to ⟨Q1⟩, this paper finds that:

§ 4.1 The effect that harder learning tasks correlate with smaller spectrally-normalized margin distributions—observed by Bartlett et al. (2017)—can be reversed by controlling normalized margin distributions.

§ 4.2 Pairs of networks can be found with *very similar* Frobenius-normalized margin distributions but *significantly different* generalization behavior.

In answer to ⟨Q2⟩, this paper finds that:

§ 5.1 In a standard learning setting, controlling normalized

[1]Caltech [2]Argo AI. Correspondence to: Alexander R. Farhang <afarhang@caltech.edu>, Jeremy Bernstein <bernstein@caltech.edu>, Yisong Yue <yyue@caltech.edu>. Code available at: https://github.com/alexfarhang/margin.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

margin does control generalization error.

Inspired by these findings, this paper further:

§ 6.1 Derives a neural network–Gaussian process (NN–GP) model of the effect of normalized margin.

§ 6.2 Finds that, in accordance with the NN–GP model, averaging the predictions of many small-normalized-margin networks improves their test error.

2. Related Work

Support vector machines. Normalized margin plays an important role in max-margin classifiers such as the support vector machine (SVM) (Cortes & Vapnik, 1995; Vapnik, 1999). SVMs minimize weight norm at fixed margin, which is equivalent to maximizing margin at fixed weight norm. Learning theoretic arguments about the SVM have worked both via VC dimension (Boser et al., 1992) and also via a PAC-Bayesian perspective (Herbrich & Graepel, 2001).

Optimization procedures. Many machine learning techniques including soft-margin SVMs, Adaboost, and logistic regression employ margin maximizing loss functions (Rosset et al., 2003). Notions of margin were often initially proposed as useful concepts for shallow models, and more recent work has extended these concepts to arbitrary layers of deep neural networks (Elsayed et al., 2018). This includes both using the entire margin distribution, or just some of its statistics (Jiang et al., 2019). Recent work has shown that, in certain problems, the gradient descent optimizer may be biased toward maximum normalized margin solutions without any explicit regularization (Soudry et al., 2018).

Generalization bounds. When bounding the risk of a learning algorithm, much of learning theory focuses on *uniform convergence bounds* that hold for the worst function in a function class. This includes both VC bounds (Vapnik, 1999) and Rademacher bounds based on spectrally-normalized margin (Bartlett et al., 2017). Another style of theoretical analysis known as PAC-Bayes theory (Langford & Shawe-Taylor, 2003; Dziugaite & Roy, 2017; Neyshabur et al., 2017) focuses on the *average* (McAllester, 1999) or *typical* (Rivasplata et al., 2020; Pérez & Louis, 2020) risk of functions in the function class.

Generalization bounds are often used to motivate *complexity measures*—meaning formulae involving network properties that are intended to measure generalization ability (Neyshabur et al., 2015; Jiang et al., 2020). A fairly comprehensive survey of generalization bounds for neural networks is provided by Pérez & Louis (2020).

Experimental studies. Researchers have found numerous puzzling empirical phenomena related to generalization in neural networks. Classic uniform convergence generalization bounds have been found to be vacuous in many realistic settings (Nagarajan & Kolter, 2019; Zhang et al., 2021a). Other effects such as *double descent* of the population risk for increasing network width are of great interest (Nakkiran et al., 2020). This has motivated a push towards greater empiricism in the study of generalization (Li et al., 2018; Mehta et al., 2021). There have also been efforts to discover complexity measures that either correlate with (Nagarajan & Kolter, 2017; Jiang et al., 2019) or cause (Jiang et al., 2020; Dziugaite et al., 2020) generalization.

3. Controlling Normalized Margin

This section defines the *normalized margin* of a neural network classification problem and develops a recipe to control this quantity. The recipe combines data normalization, a special loss function, and projected gradient descent.

3.1. Defining Normalized Margin

This subsection defines a notion of normalized margin in multi-layer perceptrons (MLPs), although the concept generalizes naturally to other network architectures.

The functional form of a depth-L MLP is given by:

$$f_L(x; w) := W_L \circ \varphi \circ W_{L-1} \circ ... \circ \varphi \circ W_1(x), \qquad (1)$$

where $x \in \mathbb{R}^{d_0}$ is the *input*, the matrices $w = (W_1, ..., W_L)$ are the *weights* and the elementwise function φ is the *nonlinearity*. This paper will restrict to the ReLU nonlinearity $\varphi(\cdot) := \max(0, \cdot)$, which is positive homogeneous. This means that the whole MLP is positive homogeneous of degree-L in the weights w and of degree-1 in the input x.

To use the network for binary classification, the output dimensionality of the network is set to 1 and the class decision is made via $x \mapsto \operatorname{sign} f_L(x; w)$. Then the margin of the network on input x with binary label $y \in \{\pm 1\}$ is given by:

$$\gamma(x, y; w) \coloneqq f_L(x; w) \cdot y.$$

To use the network for k-way classification, the output dimensionality of the network is set to k and the class decision is made via $x \mapsto \arg\max_i f_L(x; w)_i$. Then the margin of the network on input x with label $y \in \{1, ..., k\}$ is given by:

$$\gamma(x, y; w) \coloneqq f_L(x; w)_y - \max_{i \neq y} f_L(x; w)_i.$$

As defined, the margin inherits degree-L homogeneity in the weights w and degree-1 homogeneity in the input x from the MLP. This is problematic, since weight and input rescalings affect neither classification decisions nor generalization performance. Therefore, it makes sense to define the *normalized* margin $\overline{\gamma}_k$ for suitable "norms" $\|\cdot\|_\star$ and $\|\cdot\|_\dagger$:

$$\overline{\gamma}(x, y; w) \coloneqq \frac{\gamma_k(x, y; w)}{\|w\|_\star \cdot \|x\|_\dagger}.$$

The only real requirement on the "norms" is for $\|\cdot\|_\star$ and $\|\cdot\|_\dagger$ to be degree-L and degree-1 positive homogeneous respectively. This paper will consider two such choices. The first is the most *naïve*:

Definition 3.1 (Frobenius-normalized margin)**.** Let $\|\cdot\|_F$ denote the Frobenius norm, and let the lth weight matrix have dimension $d_l \times d_{l-1}$. The Frobenius normalized margin of training point (x, y) is given by:

$$\overline{\gamma}_F(x, y; w) := \gamma(x, y; w) \cdot \prod_{l=1}^{L} \frac{\sqrt{d_l}}{\|W_l\|_F} \cdot \frac{\sqrt{d_0}}{\|x\|_2}.$$

The factors of dimension d_l are included so that for standard weight and data scalings, the product term is of order one.

The second choice is a more involved notion of normalized margin that appears in a risk bound of Bartlett et al. (2017):

Definition 3.2 (Spectrally-normalized margin)**.** Let $\|\cdot\|_\sigma$ denote the spectral norm and $\|\cdot\|_{2,1}$ denote the 1-norm of the column-wise 2-norm of a matrix. The spectrally-normalized margin of training point (x, y) is given by:

$$\overline{\gamma}_\sigma(x, y; w) := \gamma(x, y; w) \cdot \frac{1}{\mathcal{R}_w} \cdot \frac{\sqrt{d_0}}{\|x\|_2},$$

where the *spectral complexity* $\mathcal{R}_w$ is defined via:

$$\mathcal{R}_w := \left(\prod_{i=l}^{L} \|W_l\|_\sigma \right) \left(\sum_{l=1}^{L} \frac{\|W_l^T - M_l^T\|_{2,1}^{2/3}}{\|W_l\|_\sigma^{2/3}} \right)^{3/2}.$$

In this expression, $m = (M_1, ..., M_L)$ are the weights of a reference network chosen before seeing the training data.

The spectral complexity $\mathcal{R}_w$ matches Equation 1.2 of Bartlett et al. (2017) after restricting to the ReLU nonlinearity, whose Lipschitz constant is one. The definition of spectrally normalized margin differs slightly in that Bartlett et al. (2017) replace the factor of $\sqrt{d_0}/\|x\|_2$ by $\|X\|_F/n$ where X is the training data matrix and n is the number of training points. When each training point is normalized separately and n is fixed—as in this paper's experiments—these definitions differ only by a constant factor.

3.2. A Recipe for Controlling Normalized Margin

In order to test the causal relationship between normalized margin and generalization, this section develops a recipe for directly controlling the distribution of Frobenius-normalized margins of a predictor over its training set (Recipe 1). Due to the mathematical relationships between different norms, this also imposes a weak form of control over the spectrally-normalized margin distribution, which is exploited in § 4.1.

The recipe has three steps: The first is to control the norm of each training input $\|x\|$. The second is to control the distribution of targeted margins $\gamma(x, y; w)$ across the training set.

Recipe 1 Controlling Frobenius-normalized margin $\overline{\gamma}_F$. The recipe targets $\overline{\gamma}_F(x_i, y_i; w) = \alpha_i$ across training points $\{x_i, y_i\}_{i=1}^n$ for an L-layer MLP $f_L(x; w)$.

① Normalize all training inputs $x \in \mathbb{R}^{d_0}$ via:

$$x \leftarrow x \cdot \frac{\sqrt{d_0}}{\|x\|_2}.$$

② Set the loss function to:

$$\mathcal{L}(w; \vec{\alpha}) \leftarrow \sum_{i=1}^{n} \left(f_L(x_i; w) - \alpha_i \cdot y_i \right)^2.$$

③ After each descent step, normalize $W_l \in \mathbb{R}^{d_l \times d_{l-1}}$:

$$W_l \leftarrow W_l \cdot \frac{\sqrt{d_l}}{\|W_l\|_F}, \text{ for layer } l = 1, ..., L.$$

And the third is to control the product of Frobenius norms of the network weights $\prod_{l=1}^{L} \|W_l\|_F$.

Step ①: controlling input norm. The norm of each input can be controlled in a data pre-processing step. In all experiments, this paper controls the norm of each input $x \in \mathbb{R}^{d_0}$ by simply projecting the input on to the hypersphere of radius $\sqrt{d_0}$. For instance, 28px × 28px MNIST images are flattened into vectors and rescaled to have a 2-norm of 28.

Step ②: controlling margin distribution. A special loss function $\mathcal{L}(w; \vec{\alpha})$ is used to target networks with either a given margin or distribution of margins over the training set:

$$\mathcal{L}(w; \vec{\alpha}) := \sum_{i=1}^{n} \left(f_L(x_i; w) - \alpha_i \cdot y_i \right)^2.$$

For binary or one-hot labels y_i, minimizing this loss function to zero corresponds to returning a network with margin $\gamma(x_i, y_i; w) = \alpha_i$ on the ith training example. Setting all α_i to the same scalar α will be referred to as *targeting margin* α. This loss function is related to the "rescaled square loss" proposed by Hui & Belkin (2021).

Step ③: controlling product of weight norms. Projected gradient descent is used to re-normalize the norm of each layer's weights after each iteration of network training. In practice, this paper employs the Nero optimizer (Liu et al., 2021), which imposes a slightly stronger form of projection than is strictly necessary. In particular, Nero enforces that each row of every weight matrix has zero sum and unit length, so that $\|W_l\|_F = \sqrt{d_l}$ as a consequence. The extra constraints are immaterial for the purposes of this paper, since the paper is only concerned with establishing examples and counterexamples of the causal effect of normalized

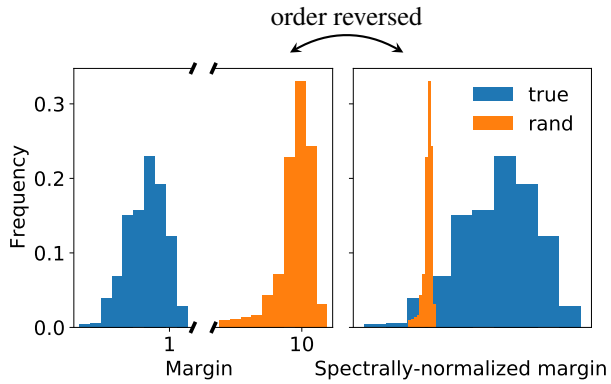

Figure 1. Reproducing the effect of Bartlett et al. (2017). *True labels* corresponds to a standard learning problem with a generalizing solution (81% test accuracy), while *random labels* corresponds to an impossible learning problem with a non-generalizing solution (8.3% test accuracy). After spectral-normalization, the margin distribution of the generalizing solution falls above that of the non-generalizing solution.

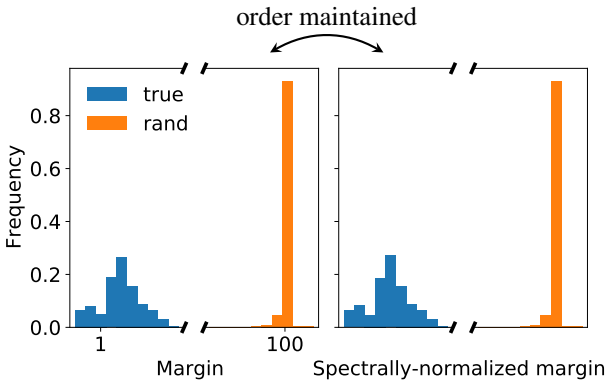

Figure 2. Breaking the effect of Bartlett et al. (2017). This figure is under the same experimental setting as Figure 1, except Recipe 1 has been used to greatly inflate the margin distribution on the *random label* task through controlled optimization. The ordering of the spectrally-normalized margin distributions no longer reflects the generalizability of the corresponding solutions (*true labels*: 81% test accuracy vs *random labels*: 10% test accuracy)

margin—it may construct those examples in any fashion.

Combining all three steps yields Recipe 1. Assuming that networks can be trained to zero loss, this recipe leads to exact control of the distribution of Frobenius-normalized margins (Definition 3.1) over the training set. Because different norms weakly control each other—for instance:

$$\|W_l\|_F / \sqrt{\min(d_l, d_{l-1})} \leq \|W_l\|_\sigma \leq \|W_l\|_F,$$

it follows that Recipe 1 also provides a weak form of control over the spectrally-normalized margin (Definition 3.2). This fact is exploited in § 4.1.

4. Normalized Margin is Insufficient to Explain Generalization

The goal of this section is to tackle ⟨**Q1**⟩: *Does normalized margin always have a causal effect on generalization?*

The main finding of the section is that normalized margin can be decoupled from generalization performance. This includes a reversal of a previously reported correspondence between spectrally-normalized margin distributions and generalization in § 4.1, and additional studies in § 4.2 that decouple Frobenius-normalized margin from generalization performance. The experiments in § 4.2 are referred to as *twin network studies* since they produce pairs of networks with very similar Frobenius-normalized margin distributions but significantly different test performance. These results constitute counterexamples suggesting that normalized margin alone cannot causally explain generalization.

4.1. Reversing Spectrally-Normalized Margin Bounds

This section shows that, through control, spectrally-normalized margin can be made to both correlate *and* anti-correlate with generalization error. These results are motivated by a prior finding that a measure of spectrally-normalized margin derived from a theoretical bound can correlate with generalization ability (Bartlett et al., 2017). This section's results highlight the risk in inferring a causal connection from a correlational study.

Background. Spectrally-normalized margin distributions have been proposed as a promising method to understand generalization in neural networks (Bartlett et al., 2017). The theory is derived from a risk bound related to Definition 3.2. In spirit, this bound is given by:

$$R(f) \lessapprox \widehat{R}_\gamma(f) + \frac{\|X\|_F \mathcal{R}_A}{\gamma n}, \qquad (2)$$

where $R(f)$ is the population risk, $\widehat{R}_\gamma(f)$ measures what Bartlett et al. (2017) refer to as the sample "ramp loss" at margin γ, $\|X\|_F$ is the Frobenius norm of the training data matrix and n is the number of training samples.

Bartlett et al. (2017) also provide a graphical way to understand the bound via the relative placement of margin distributions. In particular, for "any fixed point on the horizontal axis, if the cumulative distribution of one density is lower than the other, then it corresponds to a lower right hand side" of their bound. For the purposes of this paper, this means that if one learner's spectrally normalized margin distribution lies fully above that of a second learner, then the first learner should generalize better according to Bartlett et al. (2017)'s theory.

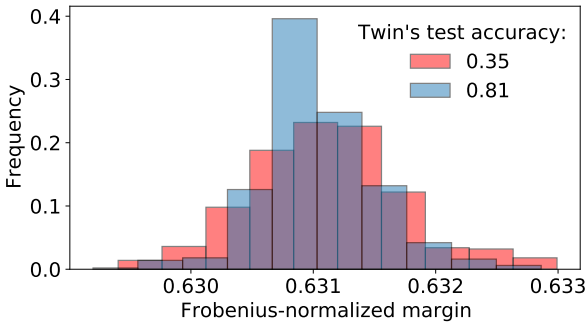

Figure 3. "Twin networks" with overlapping Frobenius-normalized margin distributions but significantly different test performance. The poorly generalizing network was selected via the attack set method of Wu et al. (2017), while its twin used standard training.

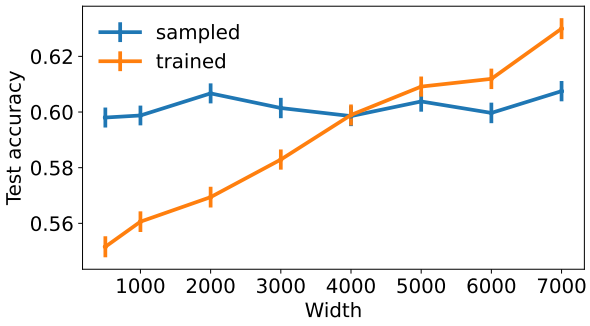

Figure 4. "Twin networks" with nearly identical Frobenius-normalized margin distributions that were selected via different optimization procedures. Each *sampled* network was found via rejection sampling, while its *trained* twin used gradient-based optimization to match its margin distribution. To make rejection sampling feasible, a very small learning problem involving 5 MNIST samples was used. At each network width, mean test accuracy and standard error of the mean is reported for 1000 pairs of twins. For near identical normalized margin distribution, the different training procedures led to different test accuracies.

Such a graphical comparison is conducted in Figure 1. In this case, spectral normalization successfully reorders two margin distributions of correctly classified points with correct or random labels such that the generalizing network's distribution places most of its mass to the right of the non-generalizing network. Given this result, it is tempting to surmise that spectrally-normalized margin may be a dominant causal factor in a network's generalization ability. The experiments in this section explore this hypothesis.

Experiments. Two sets of experiments were performed, each of which trained two MLPs on 1000 point subsets of MNIST to classify either true or randomly labeled data for 10-class classification. Using full-batch gradient-based optimization, a scaled squared loss function, and data normalization described in § 3, two networks with identical architecture were trained on either true or random labels. Note that only networks trained on true labels can possibly generalize. The experiments in this section varied the targeted label scale α, forcing the networks' margin distributions to converge to α. Networks trained on true labels always target $\alpha = 1$.

- **Experiment 1:** Spectrally-normalized margin distributions correspond with generalization ability in networks trained without weight constraints, with random labels targeting $\alpha = 10$ (Figure 1).
- **Experiment 2:** Spectrally-normalized margin distributions do not correspond with generalization ability in networks trained with Frobenius weight constraints, with random labels targeting $\alpha = 100$ (Figure 2).

Findings. The two experiments described above show that spectrally-normalized margin distributions do not track a network's generalization power. By running controlled experiments, Figures 1 and 2 show opposing correspondences between spectrally-normalized margin and generalization ability. In other words, whereas Figure 1 is consistent with the generalization bound and uncontrolled empirical study in Bartlett et al. (2017), Figure 2 shows the opposite effect. Overall, this study suggests that spectrally-normalized margin alone does not causally control generalization.

4.2. Twin Network Studies

To further explore the sufficiency of normalized margin in explaining generalization ability, this section designs two *twin network* studies to control Frobenius-normalized margins and observe their effect on generalization performance.

Background. In order to find neural network solutions with varying generalization performance, recent studies have included training points that are incorrectly labeled to reduce test performance (Wu et al., 2017; Zhang et al., 2021b). Inspired in part by these approaches, these experiments sought to produce networks that can produce similar normalized margin distributions with varying test performance through the inclusion of an *attack set* of training points with random labels (**Experiment 1**).

Previous work has studied the implicit biases of both gradient-descent (Soudry et al., 2018) and randomly sampling parameters (Valle-Perez et al., 2019; De Palma et al., 2019). In the infinite-width limit, the posterior distribution over the function space is similar between networks trained by SGD or random sampling (Mingard et al., 2021). **Experiment 2** in this section fixes normalized margin between twin networks that differ in training method (random sampling vs. gradient-based optimization) and analyzes the difference in generalization performance.

Experiment 1: Attack set twin. Twin networks were trained on identical subsets of 500 points of the MNIST

training set that targeted the same normalized margins. However, one of the networks was also trained to target the same normalized margin for additional 1000 randomly labeled train points, the attack set. Both networks are MLPs with identical architectures under Frobenius weight norm $\|w\|$, targeted margin γ, and data norm $\|X\|$ control as specified in Recipe 1.

Experiment 2: Rejection sampled twin. Pairs of networks with identical architectures (see Appendix A.3) are generated to have matching Frobenius-normalized margins on a training set, but are obtained via different optimization methods (sampling vs. gradient-based optimization). First, a network f_{sampled} is found by randomly sampling Frobenius-norm constrained networks until a small training set of binary MNIST data is perfectly classified. Then, a second twin f_{trained} is trained using Frobenius constrained gradient descent with Nero to perfectly mimic the output of f_{sampled} on the same training set, resulting in two networks with matched normalized margins. This procedure is repeated 1000 times.

Findings. Networks can have similar Frobenius-normalized margin distributions while exhibiting drastically different generalization. The results from the attack set experiments, **Experiment 1**, in Figure 3 show the Frobenius-normalized margin distributions on the correctly labeled data points each of the twins was trained on. Though the twin networks trained with (red) or without (blue) the addition of an attack set have similar normalized margin distributions, they have substantially different test performance (35% vs. 81% accuracy). The normalized margin distribution's placement can be somewhat arbitrarily controlled irrespective of generalization ability for attack set twins by targeting various margin scales α in the scaled loss function. These results suggest that neural networks could have matching normalized margin distributions and thus similar functional output on the train set, yet one could display pathologically reduced generalization.

There is also a gap in generalization performance between twin networks from **Experiment 2**, which have nearly identical normalized margins but were trained with different optimization methods. As shown in Figure 4, for MNIST 0 vs. 1 classification, some architectures (i.e some fixed widths) exhibit significantly different generalization performance between f_{sampled} and f_{trained}. This effect is observed across random seeds and different learning tasks for binary classification in MNIST and CIFAR-10 (see Appendix A.3). This difference in generalization performance between f_{sampled} and f_{trained} cannot be attributed to normalized margin, since both margin γ and weight norms $\|w\|$ are nearly identical across the two networks, and instead may be due to the implicit biases of the corresponding optimization methods.

5. Normalized Margin May Be Necessary to Explain Generalization

The goal of this section is to tackle ⟨**Q2**⟩: *Does normalized margin ever have a causal effect on generalization?*

While § 4 presented multiple settings where normalized margin does not causally impact generalization, this section seeks the opposite: settings where normalized margin does causally effect generalization.

5.1. Normalized Margin in Standard Training

This section explores the effect of normalized margin in networks trained in a more benign manner than was considered in § 4. Three experiments are conducted, each using a different subset of control presented in § 3. They all provide evidence supporting the idea that larger targeted normalized margins correspond with a network's increased test performance (bottom right panel of Figure 5).

Background. A recent study observed how a neural network's *scale of initialization* can tightly control its generalization ability (Mehta et al., 2021). In particular, by varying the scale of initialization of the first layer, one could cause a network to interpolate between good and chance test performance in the extreme case. This phenomenon may have connections with how overparameterized networks can operate in regimes known as "kernel" or "rich" depending on the model's similarity to kernel regression throughout learning (Woodworth et al., 2020; Geiger et al., 2020). But notably a network's scale of initialization can affect the scale of its weight norms and thus its normalized margin.

Experiments. Controlled experiments were designed to understand this phenomenon from the perspective of normalized margin. 2-layer MLPs were trained for 10-class classification on 1000 point subsets of MNIST. The following three experiments were run:

1. The initialization scale was varied, while the target margin was fixed to 1.
2. The targeted margin was varied, for a fixed initialization scale and with weight projection removed.
3. Frobenius-normalized margin was directly controlled and varied using Recipe 1.

Findings. These experiments reveal a correspondence between a network's generalization performance and its Frobenius-normalized margin; for a given network that can generalize, it tends to generalize better when it targets a larger normalized margin. Figure 5 demonstrates that generalization can be controlled by targeting certain normalized margins. The bottom right panel of Figure 5 registers all of the test performance curves onto the same scale, as a function of their normalized targeted margin. By constraining

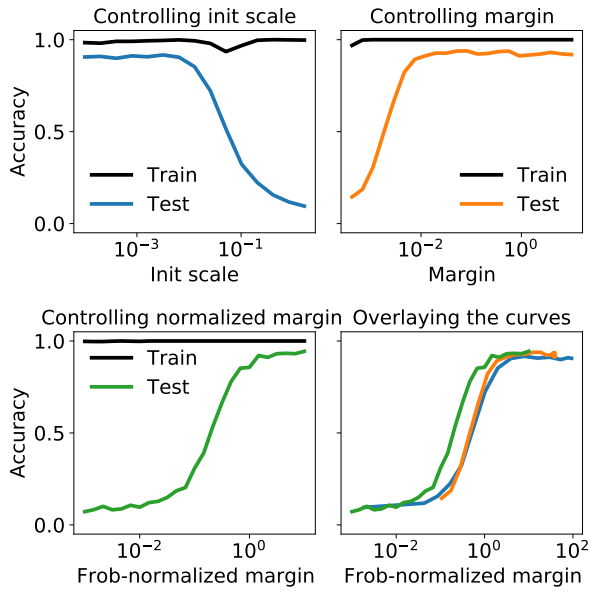

Figure 5. Accuracy as a function of controlling initialization scale, margin, or Frobenius-normalized margin. The last plot overlays the test accuracy from the other three plots as a function of each network's targeted margin divided by the layer-wise product of Frobenius norms. The close overlap suggests that Frobenius-normalized margin may largely explain the behavior in the first two plots.

the layerwise Frobenius norms and targeting specific margins with a modified squared loss function, generalization ability can be controlled by varying a network's targeted margin. Whereas the scale of initialization is set at the beginning of training and then free to vary during the dynamics of optimization, the targeted margin remains constant throughout training. These experiments suggest that in a standard training setting for networks that generalize, controlling normalized margin does control generalization.

6. Building on the Controlled Studies

So far, this paper has made two main findings. First, normalized margin seems insufficient to fully explain generalization. § 4 showed that *through careful control* one can break a reported link between normalized margin and generalization. Second, normalized margin does seem to have a strong controlling effect on generalization in less adversarial situations, as shown in § 5. The goal of this section is to develop a model that is consistent with these findings and has predictive power over the effect of normalized margin.

In particular, § 6.1 constructs a model of normalized margin based on the NN–GP correspondence. § 6.2 points out that this model makes concrete predictions about the behavior of *ensembles* of small-normalized-margin networks. These predictions are tested and verified—providing promising evidence in favor of the Gaussian process model.

6.1. A Gaussian Process Model of Normalized Margin

A Gaussian process (GP) can in principle, up to certain technical conditions, fit any function. Therefore a GP should be able to represent functions of arbitrary margin that behave arbitrarily badly away from the training data. Since GP classification is effective in practice (Rasmussen & Williams, 2005), such poorly behaving functions must not be selected for by GP inference. To test whether this is essentially the same behavior that is being observed in § 4 and § 5, one needs to build a model of GP classification that explicitly involves a normalized margin parameter.

This section accomplishes that task via the *neural network–Gaussian process correspondence* (NN-GP) (Neal, 1994; Lee et al., 2018; de G. Matthews et al., 2018). Consider an L-layer ReLU-MLP (Equation 1) with weight matrices $w = (W_1, ..., W_L)$, where the lth layer's weight matrix W_l has dimension $d_l \times d_{l-1}$. Consider randomly sampling the weights at each layer according to $W_l^{(ij)} \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2/d_{l-1})$, where the parameter σ sets the prior scale of each layer.

Sending the layer widths $d_1, ..., d_L$ to infinity, the NN–GP correspondence states that the distribution of network outputs under this prior on the weights is given by a GP with mean zero and covariance function (Lee et al., 2018):

$$\Sigma(x, x') = \sigma^{2L} \cdot \underbrace{h \circ ... \circ h}_{L-1 \text{ times}} \left(\frac{x^T x'}{d_0} \right),$$

where $h(t) := \frac{1}{\pi} \cdot [\sqrt{1-t^2} + t \cdot (\pi - \arccos t)]$. This is the *compositional arccosine kernel* of Cho & Saul (2009).

To construct the posterior distribution over a test point x given a training set $X = \{x_1, ..., x_n\}$ and a vector of binary training labels $Y \in \{\pm 1\}^n$, one requires the *Gram matrix* Σ_{XX}, *Gram vector* Σ_{xX} and *Gram scalar* Σ_{xx} defined by:

$$\Sigma_{XX}^{(ij)} := \Sigma(x_i, x_j);\ \Sigma_{xX}^{(i)} := \Sigma(x, x_i);\ \Sigma_{xx} := \Sigma(x, x).$$

This paper also defines the *normalized Gram tensors* via:

$$\widehat{\Sigma}_{XX} := \frac{\Sigma_{XX}}{\sigma^{2L}}; \qquad \widehat{\Sigma}_{xX} := \frac{\Sigma_{xX}}{\sigma^{2L}}; \qquad \widehat{\Sigma}_{xx} := \frac{\Sigma_{xx}}{\sigma^{2L}}.$$

Consider scaling up the training labels by a *margin parameter* γ. The distribution over functions that interpolate the training points $(X, \gamma Y)$ evaluated at test point x is then:

$$\begin{aligned} &\mathcal{N}\left(\gamma \cdot \Sigma_{xX}\Sigma_{XX}^{-1}Y, \Sigma_{xx} - \Sigma_{xX}\Sigma_{XX}^{-1}\Sigma_{Xx}\right) \\ &= \mathcal{N}\big(\gamma \cdot \underbrace{\widehat{\Sigma}_{xX}\widehat{\Sigma}_{XX}^{-1}Y}_{=:C_1}, \sigma^{2L} \cdot \underbrace{\widehat{\Sigma}_{xx} - \widehat{\Sigma}_{xX}\widehat{\Sigma}_{XX}^{-1}\widehat{\Sigma}_{Xx}}_{=:C_2}\big). \end{aligned}$$

The signal-to-noise ratio of this GP posterior is set by the parameter γ/σ^L. Since γ sets the scale of the outputs, and σ sets the prior scale of each layer's weights, γ/σ^L has an interpretation as the *normalized margin* of the posterior. The next section tests the effect of γ/σ^L on generalization.

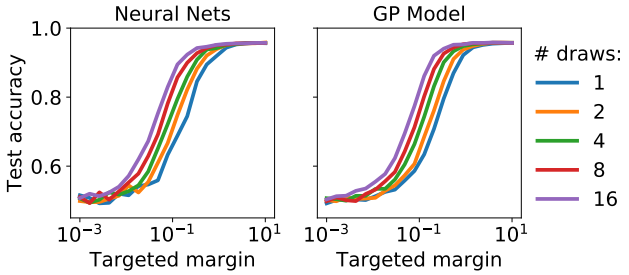

Figure 6. Averaging the predictions of many NNs (left) and NN–GP posterior samples (right) as a function of normalized margin. For NNs, *# draws* refers to the number of networks trained by Nero. *Targeted margin* refers to *Frobenius-normalized margin* for NNs and γ/σ^L (see Equation 3) for NN–GP draws. Test performance increases in a very similar way for both NNs and NN–GP draws, as a function of both ensemble size and normalized margin.

6.2. Ensemble Behavior of Normalized Margin

This section studies the NN–GP model of normalized margin developed in § 6.1. The central prediction of the model is shown to be that *averaging small-normalized-margin functions* should have the same effect on generalization as *increasing the normalized margin*. This effect is found to map back to finite width MLPs in Figure 6.

Background. In § 6.1, it was shown that the NN–GP predictive distribution at target margin γ and layer scale σ is:

$$\mathcal{N}(\gamma \cdot C_1, \sigma^{2L} \cdot C_2),$$

where C_1 and C_2 are independent of γ and σ. Also, the average of m iid draws from this distribution follows:

$$\mathcal{N}(\gamma \cdot C_1, \sigma^{2L} \cdot C_2/m). \tag{3}$$

Since the variance of the posterior corresponds to adding Gaussian noise to the predictions, it is only reasonable that this variance should harm prediction quality. To force the posterior to concentrate on its mean, one may either:

a) Let the normalized margin $\gamma/\sigma^L \to \infty$;

b) Let the number of ensemble members $m \to \infty$.

This model may be tested for finite width NNs simply by replacing γ/σ^L with the Frobenius-normalized margin.

Experiments. Finite width MLPs were trained on subsets of MNIST for even/odd classification using layerwise Frobenius control and margin control as prescribed in Recipe 1. Each individual model in a given ensemble is trained on the same subset of data and their output activations are averaged and then binarized to form a prediction. This is performed over a range of targeted Frobenius-normalized margins. The same experiment was repeated for NN–GP draws using the ensembled predictive distribution given in Equation 3.

Findings. For both ensembles of networks and GP draws, test performances increases as a function of both ensemble size and targeted normalized margin. As shown in Figure 6, individual large-margin classifiers attain the same test accuracy as an ensemble-average of small-margin classifiers. The functional form of the curves for finite width NNs and NN–GP draws is remarkably similar.

7. Discussion

The paper has presented a set of controlled experiments to address the sufficiency of normalized margin to explain generalization in *all settings*, and its necessity in *typical settings*. The counterexamples in § 4 show that spectrally- and Frobenius-normalized margin are not sufficient to explain generalization performance in general, since for instance normalized margin distributions can be somewhat arbitrarily inflated through controlled optimization without yielding good generalization performance. However, the positive examples in § 5 demonstrate that normalized margin can control test performance in less adversarial settings.

This section discusses three topics: first, how the paper relates to the pursuit of a more scientific understanding of deep learning; second, the potential for the paper's results and techniques to inform future developments in learning theory; and third, a possible application of normalized margin control to uncertainty quantification via deep ensembles.

7.1. Predictive Models and Controlled Studies

At the core of the scientific process is the construction of predictive theories and models, which are in turn used to generate and test falsifiable hypotheses via experiment. Controlled studies are often considered the "gold standard" in experimental design for this kind of hypothesis testing. Under this light, this paper has developed a means of controlling normalized margin in order to test a generalization theory based on spectrally normalized margin distributions (§ 4.1). Finding this theory wanting, the paper constructed a new model based on a notion of normalized margin in Gaussian processes (§ 6.1). This new model was found to yield accurate predictions about the behavior of ensembles of neural networks.

A model is a simplification or abstraction of a system that throws away the messy details while attempting to capture the system's essence. Models are important for their ability to reveal insights and relationships about a system that are difficult to see directly. The NN–GP model of normalized margin proposed in § 6.1 suggests that there are still valuable insights that can be drawn from established models. In this instance, the NN–GP model can provide insight about generalization in neural networks by focusing on the function space prior without appealing to more involved models such as the *neural tangent kernel* (Jacot et al., 2018).

In contrast to the type of controlled experimentation con-

ducted in this paper, much work studies phenomena in deep learning without experimental intervention on the objects of theoretical interest (Bartlett et al., 2017). Other work moves toward a greater level of control—for instance Jiang et al. (2020) control optimization hyperparameters and subsequently attempt to tease out causal relationships between complexity measures and generalization. However, this paper goes a step further by directly intervening on the quantities of theoretical interest. This style of controlled experimentation—which has appeared in other areas of machine learning research (Balakrishnan et al., 2020)—might facilitate a richer feedback loop with theory in pursuit of a more complete understanding of generalization.

7.2. Implications for Learning Theory

This paper used a controlled study to find a counterexample to a hypothesized causal relationship about generalization in neural networks. This technique could be used in a more positive sense to design improved generalization theories. For instance, one popular framework for generalization theory known as *uniform convergence* derives risk bounds that hold for all classifiers within a specified structural family. Controlled investigation of generalization in different structural families could lead to the discovery of new structural families for uniform convergence theory that are immune to the kind of counterexamples witnessed in this paper.

Controlled experiments may also help in studying other generalization theories such as *PAC-Bayes theory* (McAllester, 1999). PAC-Bayes bounds hold for distributions of classifiers, and controlled studies might enable more efficient investigation of special distributions of classifiers. A concrete example of this is the experiment in Figure 6, where controlled optimization enables the study of distributions of classifiers conditioned on a prescribed normalized margin.

7.3. Implications for Uncertainty Quantification

The techniques developed in this paper may have applications beyond learning theory. In uncertainty quantification, the challenge is to coax a machine learning model into reporting a meaningful notion of confidence in its predictions. One popular technique, known as *deep ensembles* (Lakshminarayanan et al., 2017), involves training many neural networks with different random seeds in order to obtain a spread of predictions. But according to the Gaussian process model of normalized margin developed in Section 6.1, if one is not careful and trains each deep ensemble member to large normalized margin, the trained ensemble members may collapse on to the same function. This model would suggest that to obtain accurate uncertainty information from a deep ensemble, each ensemble member should be trained to small normalized margin. As such, normalized margin control may play a role in uncertainty quantification.

Acknowledgements

The authors are grateful to the anonymous reviewers for their helpful comments. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-1745301. This material was also supported by the following grants: NSF #1918865; ONR N00014-21-1-2483.

References

Balakrishnan, G., Xiong, Y., Xia, W., and Perona, P. Towards causal benchmarking of bias in face analysis algorithms. In *European Conference on Computer Vision*, 2020.

Bartlett, P. L., Foster, D. J., and Telgarsky, M. J. Spectrally-normalized margin bounds for neural networks. In *Neural Information Processing Systems*, 2017.

Boser, B. E., Guyon, I. M., and Vapnik, V. N. A training algorithm for optimal margin classifiers. In *Workshop on Computational Learning Theory*, 1992.

Cho, Y. and Saul, L. Kernel methods for deep learning. In *Neural Information Processing Systems*, 2009.

Cortes, C. and Vapnik, V. Support-vector networks. *Machine Learning*, 1995.

de G. Matthews, A. G., Hron, J., Rowland, M., Turner, R. E., and Ghahramani, Z. Gaussian process behaviour in wide deep neural networks. In *International Conference on Learning Representations*, 2018.

De Palma, G., Kiani, B., and Lloyd, S. Random deep neural networks are biased towards simple functions. *Neural Information Processing Systems*, 2019.

Dziugaite, G. K. and Roy, D. M. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. In *Uncertainty in Artificial Intelligence*, 2017.

Dziugaite, G. K., Drouin, A., Neal, B., Rajkumar, N., Caballero, E., Wang, L., Mitliagkas, I., and Roy, D. M. In search of robust measures of generalization. In *Neural Information Processing Systems*, 2020.

Elsayed, G. F., Krishnan, D., Mobahi, H., Regan, K., and Bengio, S. Large margin deep networks for classification. In *Neural Information Processing Systems*, 2018.

Geiger, M., Spigler, S., Jacot, A., and Wyart, M. Disentangling feature and lazy training in deep neural networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2020.

Herbrich, R. and Graepel, T. A PAC-Bayesian margin bound for linear classifiers: Why SVMs work. In *Neural Information Processing Systems*, 2001.

Hui, L. and Belkin, M. Evaluation of neural architectures trained with square loss vs. cross-entropy in classification tasks. In *International Conference on Learning Representations*, 2021.

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. In *Neural Information Processing Systems*, 2018.

Jiang, Y., Krishnan, D., Mobahi, H., and Bengio, S. Predicting the generalization gap in deep networks with margin distributions. In *International Conference on Learning Representations*, 2019.

Jiang, Y., Neyshabur, B., Mobahi, H., Krishnan, D., and Bengio, S. Fantastic generalization measures and where to find them. In *International Conference on Learning Representations*, 2020.

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. In *International Conference on Learning Representations*, 2017.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Neural Information Processing Systems*, 2017.

Langford, J. and Shawe-Taylor, J. PAC-Bayes & margins. *Neural Information Processing Systems*, 2003.

Lee, J., Sohl-Dickstein, J., Pennington, J., Novak, R., Schoenholz, S., and Bahri, Y. Deep neural networks as Gaussian processes. In *International Conference on Learning Representations*, 2018.

Li, H., Xu, Z., Taylor, G., Studer, C., and Goldstein, T. Visualizing the loss landscape of neural nets. In *Neural Information Processing Systems*, 2018.

Liu, Y., Bernstein, J., Meister, M., and Yue, Y. Learning by turning: Neural architecture aware optimisation. In *International Conference on Machine Learning*, 2021.

McAllester, D. A. Some PAC-Bayesian theorems. *Machine Learning*, 1999.

Mehta, H., Cutkosky, A., and Neyshabur, B. Extreme memorization via scale of initialization. In *International Conference on Learning Representations*, 2021.

Mingard, C., Valle-Pérez, G., Skalse, J., and Louis, A. A. Is SGD a Bayesian sampler? Well, almost. *Journal of Machine Learning Research*, 2021.

Nagarajan, V. and Kolter, J. Z. Generalization in deep networks: The role of distance from initialization. In *NeurIPS Workshop on Deep Learning: Bridging Theory and Practice*, 2017.

Nagarajan, V. and Kolter, J. Z. Uniform convergence may be unable to explain generalization in deep learning. In *Neural Information Processing Systems*, 2019.

Nakkiran, P., Kaplun, G., Bansal, Y., Yang, T., Barak, B., and Sutskever, I. Deep double descent: Where bigger models and more data hurt. In *International Conference on Learning Representations*, 2020.

Neal, R. M. *Bayesian Learning for Neural Networks*. Ph.D. thesis, Department of Computer Science, University of Toronto, 1994.

Neyshabur, B., Tomioka, R., and Srebro, N. Norm-based capacity control in neural networks. In *Conference on Learning Theory*, 2015.

Neyshabur, B., Bhojanapalli, S., and Srebro, N. A PAC-Bayesian approach to spectrally-normalized margin bounds for neural networks. *International Conference on Learning Representations*, 2017.

Pérez, G. V. and Louis, A. A. Generalization bounds for deep learning. *arXiv:2012.04115*, 2020.

Rasmussen, C. E. and Williams, C. K. I. *Gaussian Processes for Machine Learning*. MIT Press, 2005.

Rivasplata, O., Kuzborskij, I., Szepesvári, C., and Shawe-Taylor, J. PAC-Bayes analysis beyond the usual bounds. In *Neural Information Processing Systems*, 2020.

Rosset, S., Zhu, J., and Hastie, T. Margin maximizing loss functions. In *Neural Information Processing Systems*, 2003.

Soudry, D., Hoffer, E., Nacson, M. S., Gunasekar, S., and Srebro, N. The implicit bias of gradient descent on separable data. *Journal of Machine Learning Research*, 2018.

Valle-Perez, G., Camargo, C. Q., and Louis, A. A. Deep learning generalizes because the parameter–function map is biased towards simple functions. In *International Conference on Learning Representations*, 2019.

Vapnik, V. *The Nature of Statistical Learning Theory*. Springer, 1999.

Woodworth, B., Gunasekar, S., Lee, J. D., Moroshko, E., Savarese, P., Golan, I., Soudry, D., and Srebro, N. Kernel and rich regimes in overparametrized models. In *Conference on Learning Theory*, 2020.

Wu, L., Zhu, Z., and Weinan, E. Towards understanding generalization of deep learning: Perspective of loss landscapes. In *ICML Workshop on Principled Approaches to Deep Learning*, 2017.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 2021a.

Zhang, S., Reid, I., Pérez, G. V., and Louis, A. Why flatness does and does not correlate with generalization for deep neural networks. *arXiv:2103.06219*, 2021b.

A. Experimental Details

A.1. Reversing Spectrally Normalized Margin Bounds

Depth 5, width 5000 fully connected neural networks were trained for 10-class classification on subsets of 1000 training points from MNIST and evaluated on the entire MNIST test set. Rectified Linear unit (ReLU) activations were used throughout all experiments. The data was normalized according to Recipe 1 without the Frobenius control. The networks generating the margin distributions in Figure 1 were trained with a label-scaled squared loss function (true data label scale: 1, random data label scale: 10), full batch gradient descent with a learning rate of 0.01 and and an exponential learning rate decay of 0.999. They were trained to within 95% training accuracy. The networks in Figure 1 had test performance of 88% for the correctly labeled network and 8.3% for the randomly labeled network.

The networks generating the margin distributions in Figure 2 had identical architectures as above, but were trained with Frobenius control using the Nero optimizer (learning rate: 0.01, Nero β: 0.999) to perfect classification accuracy. Targeted label scales were set to 1 (true data) and 100 (random labels). Spectral-normalization was calculated with respect to the weights at initialization and included $\|X\|$ and n correction. The networks in this figure had test performance of 81% for the correctly labeled network and 10% for the randomly labeled network.

A.2. Twin Network Study: Attack Set

Two layer fully connected neural networks (width: 2048) were trained using Frobenius control with Nero (learning rate: 0.01, β: 0.99997, 100,000 epochs) to fit 500 training points from MNIST for 10-class classification. One network's training set was further augmented by adding the attack set of the 1000 more train points labeled randomly. They were both evaluated on the correctly labeled 10,000 test points and achieved perfect classification accuracy. Only the margins for the correctly labeled 500 training points are presented in Figure 3. Figure 7 shows accuracy of twin networks that have (attack) or have not (control) been trained on an attack set as a function of targeted normalized margin.

A.3. Twin Network Study: Optimization Dependence

For MNIST experiments, the architecture was a depth 7 MLP with ReLU activation. For CIFAR-10 experiments, the architecture consisted of 3 convolutional layers, followed by a flatten, followed by 3 linear layers. For MNIST experiments, the intermediate layer widths used were: 500, 1000, 2000, 3000, 4000, 5000, 6000, 7000. For CIFAR-10 experiments, the width of the linear layers were fixed to be 500, and the channel width of the convolutional layers varied as per the following list: 3, 5, 10, 20, 40, 80, 160, 320. When performing random sampling of parameters, weights were drawn from a $\mathcal{N}(0, 1)$ distribution.

To ensure the chosen architectures could capture their binary classification task, all architectures were trained on their respective binary classification task for 50 epochs. For MNIST 0 vs. 1 classification, the training set size was 12665 and test set size was 2115. For MNIST 4 vs. 7 classification, the training set size was 12107 and test size was 2010. For MNIST 3 vs. 8 classification, the training set size was 11982 and test set size was 1984. For CIFAR-10 dog vs. ship, the training set size was 10000 and test set size was 2000. On MNIST binary classification tasks (0 vs. 1, 3 vs. 8, and 4 vs. 7), the worst training accuracy across all architectures at the end of training was 100%; for CIFAR-10, the worst training accuracy across all architectures at the end of training was 73.81%.

As noted in section 4.2, networks were trained to match the margin of sampled networks. Training used a loss threshold of 0.000001, which indicates that $||f_{\text{sampled}}(x) - f_{\text{trained}}(x)||_2 < 0.000001$ for the given set of training examples x. To justify this choice, training loss was inspected relative to the scale of the margin of f_{sampled} for each pair $(f_{\text{trained}}, f_{\text{sampled}})$. Table 1 reports the worst relative error across all architectures and seeds for a corresponding binary classification task. The worst relative error is very small, indicating that there is a negligible difference in margin between f_{sampled} and f_{trained}.

A.4. Normalized Margin in Standard Training

2-Layer neural networks were trained to fit 1000 point subsets of MNIST and evaluated on the whole test set. They were trained using either full batch gradient descent or full batch Nero (β: 0.999) while varying the initialization scale or targeted margin scale. Networks were trained between 50,000 to 250,000 epochs (learning rates between 0.9998 and 0.999998) to achieve training accuracy marked in Figure 5. Figure 10 shows accuracy as a function of Frobenius normalized targeted margin for networks trained on true or random data.

A.5. Ensemble Behavior of Normalized Margin

Depth 5, width 2048 MLPs were trained on 1000 samples of MNIST digits, to perform even/odd classification. Networks were trained using full-batch Nero with initial learning rate 0.01, beta set to 0.999 and learning rate decay factor 0.99 per iteration. Networks were trained for 500 iterations. A variety of margins were targeted ranging from 10^{-3} up to 10^{1}. The experiment was repeated for Gaussian process draws using the predictive given in Equation 3 with C_1 and C_2 defined earlier in that section.

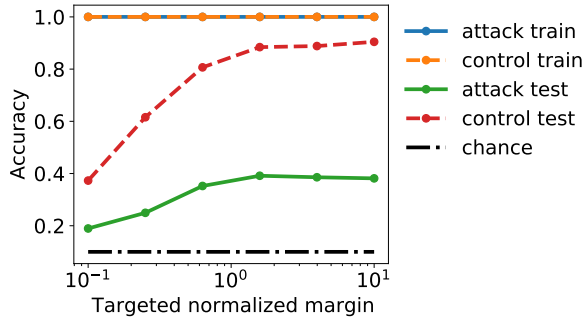

Figure 7. Test and training accuracy of twin networks with or without addition of an attack set as a function of targeted Frobenius normalized margin.

Learning task	Worst relative error
MNIST: 0 vs. 1	0.0000355
MNIST: 3 vs. 8	0.0000568
MNIST: 4 vs. 7	0.0000248
CIFAR-10: dog vs. ship	0.0002589

Table 1. Worst relative error between randomly sampled and gradient-descent trained networks across all 1000 samples and seeds, for a given learning task.

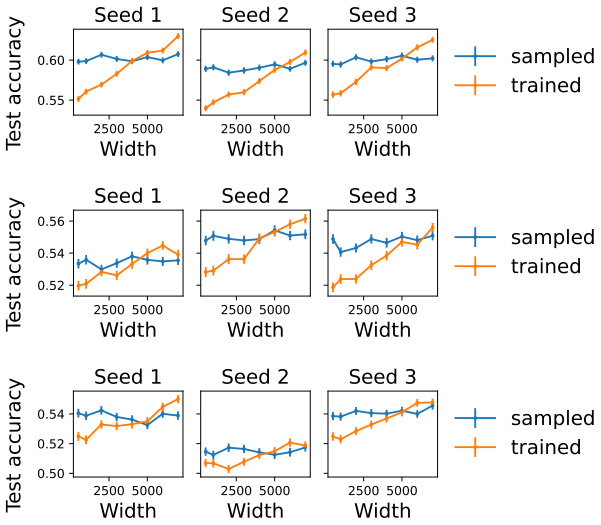

Figure 8. Test performance of f_{sampled} vs f_{trained} for 0 vs. 1 (top), 4 vs. 7. (middle), and 3 vs. 8 (bottom) MNIST binary classification task. For each width, the mean test accuracy and standard error bars are presented for 1000 pairs of (f_{sampled}, f_{trained}).

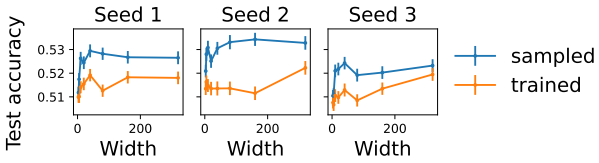

Figure 9. Test performance of f_{sampled} vs f_{trained} for dog vs. ship CIFAR-10 binary classification task. For each width, the mean test accuracy and standard error bars are presented for 1000 pairs of (f_{sampled}, f_{trained}).

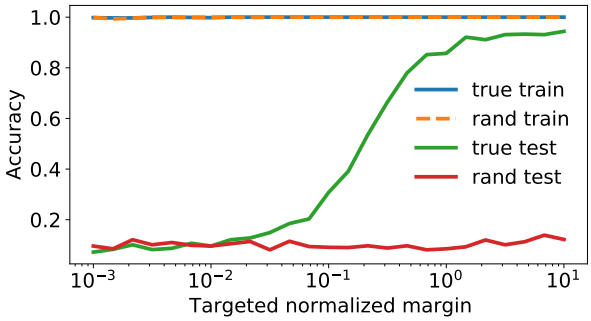

Figure 10. Train and test performance of networks trained to target specified Frobenius-normalized margins.