Simplicity Bias and the Robustness of Neural Networks

Harshay Shah [1] **Kaustav Tamuly** [1] **Aditi Raghunathan** [2] **Prateek Jain** [1] **Praneeth Netrapalli** [1]

Abstract

We identify *Simplicity Bias* (SB) —the tendency of standard training procedures such as Stochastic Gradient Descent (SGD) to find simple models— as a unifying explanation for the nonrobustness of neural networks in out-of-distribution (OOD) and adversarial settings. To systematically understand the effect of SB on robustness, we introduce a collection of piecewise-linear and image-based datasets that (a) naturally incorporate a precise notion of simplicity and (b) capture the subtleties of neural networks trained on real datasets. Through theory and experiments on these datasets, we show that SB of SGD and variants is extreme: neural networks tend to rely exclusively on the simplest feature and remain invariant to all predictive complex features. Consequently, the extreme nature of SB explains why seemingly benign distribution shifts and small adversarial perturbations significantly degrade model performance.

1. Introduction

Several works have proposed Simplicity Bias (SB)—the tendency of standard training procedures such as Stochastic Gradient Descent (SGD) to find simple models—to justify why neural networks generalize well (Arpit et al., 2017; Nakkiran et al., 2019; Valle-Perez et al., 2019). However, the precise notion of simplicity remains vague. Furthermore, previous settings (Soudry et al., 2018; Gunasekar et al., 2018) that use SB to justify why neural networks generalize well do not simultaneously capture the brittleness of neural networks—a widely observed phenomenon in practice (Szegedy et al., 2013; Jo & Bengio, 2017).

Our goal is to formally understand and probe the *simplicity bias* (SB) of neural networks in a setting that is rich enough to capture real-world problems and, at the same time, amenable to theoretical analysis & controlled experiments. Our starting point is the observation that on real-world datasets, there are several distinct ways to discriminate between labels (e.g., by inferring shape, color etc. in image classification) that define decision boundaries of varying complexity. For example, in the image classification task of white swans vs. black bears, i) a linear-like (simple) classifier that only looks at color, as well as ii) a nonlinear (complex) classifier that infers shape, would both have perfect predictive power. To systematically understand SB, we design synthetic and image-based datasets wherein different coordinates / blocks define decision boundaries of varying complexity. We refer to each coordinate / block as a *feature* and define a precise notion of feature *simplicity* based on the *simplicity of the corresponding decision boundary*.

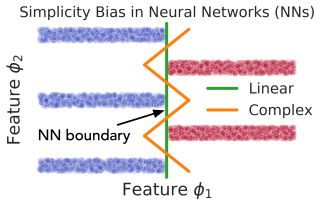


Figure 1: Simple vs. complex features

Proposed dataset. Figure 1 illustrates a stylized version of the proposed synthetic dataset with two features, ϕ_1 and ϕ_2, that can perfectly predict the label with 100% accuracy, but differ in simplicity. The simplicity of a feature is precisely determined by the *minimum* number of linear pieces in the deci sion boundary that achieves optimal classification accuracy using that feature. For example, in Figure 1, the simple feature ϕ_1 requires a linear decision boundary to perfectly predict the label, whereas complex feature ϕ_2 requires four linear pieces. Along similar lines, we also introduce a collection of image-based datasets in which each image concatenates `MNIST` images (simple feature) and `CIFAR-10` images (complex feature). The proposed datasets, which incorporate features of varying predictive power and simplicity, allow us to systematically investigate and measure SB in trained neural networks.

Observations from new dataset. The ideal decision boundary that achieves high accuracy *and* robustness relies on all features to obtain a large margin (minimum distance from any point to decision boundary). For example, the orange decision boundary in Figure 1 that learns ϕ_1 and ϕ_2 attains 100% accuracy and is more robust to perturbations than the linear boundary because of larger margin. Given the expressive power of neural networks, one might expect that a network trained on the dataset in Figure 1 would result in the larger-margin orange piecewise-linear boundary. However, we find quite the opposite: trained neural net-

[1]Microsoft Research India [2]Stanford University. Correspondence to: Harshay Shah <harshay.rshah@gmail.com>.

Presented at the ICML 2020 Workshop on Uncertainty and Robustness in Deep Learning.

works have a linear boundary. Surprisingly, neural networks exclusively learn the simpler feature ϕ_1 and remain *completely invariant* to ϕ_2. More generally, we observe that SB is extreme: models fully ignore complex predictive features in the presence of simple predictive features.

Implications of extreme SB. Theoretical analysis and controlled experiments on the proposed synthetic and image-based datasets reveal two major pitfalls of extreme SB:
(i) Lack of robustness: Neural networks (NNs) exclusively latch on to the simplest feature (e.g., background) at the expense of very small margin and completely ignore complex predictive features (e.g., semantics of the object). In Section 3, we show that this phenomenon (a) results in susceptibility to small adversarial perturbations and spurious correlations and (b) explains the existence of data-agnostic and model-agnostic universal adversarial perturbations (Moosavi-Dezfooli et al., 2017) observed in practice.
(ii) Lack of reliable confidence estimates: Ideally, a network should have high confidence only if all predictive features agree in their prediction. However, due to extreme SB, models have high confidence even if all complex predictive features are swapped across classes, thereby resulting in inaccurate, high-confidence estimates (Guo et al., 2017).

Comparison to related work: Multiple works in literature (a) differentially characterize *learned* features and *desired* features—statistical regularities vs. high-level concepts (Jo & Bengio, 2017), syntactic cues vs. semantic meaning (McCoy et al., 2019), robust vs. non-robust features (Ilyas et al., 2019)—and (b) posit that the mismatch between these features results in non-robustness. Our work instead probes *why* neural networks prefer one set of features over another and unifies the aforementioned disparate feature characterizations through the lens of feature simplicity.

2. Preliminaries: Setup and Metrics

Setting and metrics: We focus on binary classification. Given samples $\widehat{\mathcal{D}} = \{(x_i, y_i)\}_{i=1}^n$ from distribution $\mathcal{D}$ over $\mathbb{R}^d \times \{-1, 1\}$, the goal is to learn a scoring function $s(x) : \mathbb{R}^d \to \mathbb{R}$ (such as logits), and an associated classifier $f : \mathbb{R}^d \to \{-1, 1\}$ defined as $f(x) = 2h(x)-1$ where $h(x) = \mathbb{1}\{\text{softmax}(s(x)) < 0.5\}$. In addition to standard and δ-robust accuracies widely studied in literature, we introduce two metrics that quantitatively capture the extent to which a model relies on different input coordinates (or features). Let S denote some subset of coordinates $[d]$ and $\overline{\mathcal{D}}^S$ denote the S-randomized distribution, which is obtained as follows: given $\mathcal{D}^S$, the marginal distribution of S, $\overline{\mathcal{D}}^S$ independently samples $((x^S, x^{S^c}), y) \sim \mathcal{D}$ and $\overline{x}^S \sim \mathcal{D}^S$ and then outputs $((\overline{x}^S, x^{S^c}), y)$. In $\overline{\mathcal{D}}^S$, the coordinates in S are rendered independent of the label y. The two metrics are as follows.

Definition 1 (Randomized accuracy)**.** Given data distribution $\mathcal{D}$, and subset of coordinates $S \subseteq [d]$, the S-randomized accuracy of a classifier f is given by: $\mathbb{E}_{\overline{\mathcal{D}}^S}\left[\mathbb{1}_{\{f(x)=y\}}\right]$.

Definition 2 (Randomized AUC)**.** Given distribution $\mathcal{D}$ and coordinates subset $S \subseteq [d]$, S-randomized AUC of f is the area under the precision-recall curve of distribution $\overline{\mathcal{D}}^S$.

	Accuracy	AUC	Logits
S-Randomized	0.5	0.5	randomly shuffled
S^c-Randomized	standard accuracy	standard AUC	essentially identical

Table 1: If the randomized metrics of a model are as above, then the model relies *exclusively* on S and is *invariant* to S^c.

Our experiments use {S, S^c }-randomized metrics—accuracy, AUC, logits—to establish that f depends *exclusively on some features* S and *remains invariant to the rest* S^c. First, if (a) S-randomized accuracy and AUC equal 0.5 and (b) S-randomized logit distribution is a random shuffling of the original distribution (i.e., logits in the original distribution are randomly shuffled across true positives and true negatives), then f depends *exclusively* on S. Conversely, if (a) S^c-randomized accuracy and AUC are equal to standard accuracy and AUC and (b) S^c-randomized logit distribution is essentially identical to the original distribution, then f is *invariant* to S^c. Table 1 summarizes these observations.

2.1. Datasets

One-dimensional Building Blocks: We use two one-dimensional data blocks, linear and k-slabs, as shown in the top row of Figure 2. In **linear** blocks, positive and negative examples are uniformly distributed in $[0.1, 1]$ and $[-1, -0.1]$ respectively. In k**-slab** blocks, positive and negative examples are placed in k alternating, well-separated intervals.

Simplicity of Building Blocks: Linear classifiers can attain the optimal (Bayes) accuracy of 1 on the linear block. For k-slabs, however, $(k-1)$-piecewise linear classifiers are required to obtain the optimal accuracy of 1. Consequently, the building blocks have a natural notion of simplicity: *minimum number of pieces required by a piecewise linear classifier to attain optimal accuracy*. With this notion, the linear block is simpler than k-slab blocks when $k > 2$, and k-slab blocks are simpler than ℓ-slab blocks when $k < \ell$.

Multi-dimensional Synthetic Datasets: We use two d-dimensional datasets, as shown in Figure 2, wherein each coordinate corresponds to one of two building blocks.

- `LMS-k`: Linear and multiple k-slabs; the first coordinate is a linear block and the remaining $d-1$ coordinates are independent k-slab blocks.
- `MS-(5,7)`: 5-slab and multiple 7-slab blocks; the first coordinate is a 5-slab block and the remaining $d-1$ coordinates are independent 7-slab blocks, as shown in Figure 2.

We now describe the `LSN` (linear, 3-slab & noise) dataset, a stylized version of `LMS-k` that is amenable to theoretical analysis. In `LSN`, conditioned on the label y, the first and second coordinates of x are *singleton* linear and 3-slab blocks: linear and 3-slab blocks have support on $\{-1, 1\}$ and $\{-1, 0, 1\}$ respectively. The remaining coordinates are standard gaussians and not predictive of the label. Note that (a) all synthetic data comprise features of equal predic-

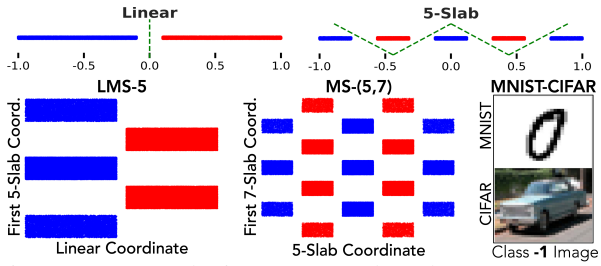


Figure 2: (Synthetic & Image-based Datasets) One-dimensional linear and k-slab blocks (top row) are used to construct two multi-dimensional datasets (bottom row): `LMS-5` (Linear & Multiple 5-Slabs) and `MS-(5,7)` (5-Slab and Multiple 7-Slabs). The `MNIST-CIFAR` data vertically concatenates `MNIST` and `CIFAR` images.

tive power but varying simplicity and (b) the d-dim. data ($d = 50$ by default) can be *perfectly* classified.

MNIST-CIFAR Data: The `MNIST-CIFAR` dataset consists of two classes: images in class -1 and class 1 are vertical concatenations of `MNIST` digit zero & `CIFAR-10` automobile and `MNIST` digit one & `CIFAR-10` truck images respectively, as shown in Figure 2. The training and test datasets comprise 50,000 and 10,000 images of size $3 \times 64 \times 32$. The `MNIST-CIFAR` dataset mirrors the structure in the synthetic `LMS-k` dataset— the `MNIST` and `CIFAR` blocks correspond to the simpler linear and more complex k-slab blocks in `LMS-k` respectively.

3. Extreme SB Leads to Non-Robustness

We first establish the *extreme* nature of SB. *If all features have full predictive power, NNs rely exclusively on the simplest feature* $\mathtt{S}$ *and remain invariant to all complex features* $\mathtt{S}^c$. Then, we explain why extreme SB results in poor OOD performance and adversarial vulnerability

3.1. Neural networks provably exhibit Simplicity Bias

Theorem 1 proves that one-hidden-layer ReLU NNs trained with mini-batch gradient descent (GD) on the `LSN` dataset (described in Section 2.1) provably learns a classifier that *exclusively* relies on the "simple" linear coordinate, thus exhibiting simplicity bias at the cost of margin. That is, the model does not learn the larger-margin classifier that relies on the linear *and* slab coordinates (recall Figure 1). The proof of Theorem 1 is presented in the appendix.

Theorem 1. *Let* $f(x) = \sum_{j=1}^{k} v_j \cdot ReLU(\sum_{i=1}^{d} w_{i,j} x_i)$ *denote a one-hidden-layer neural network with* k *hidden units and ReLU activations. Set* $v_j = \pm 1/\sqrt{k}$ *w.p.* $1/2$ $\forall j \in [k]$. *Let* $\{(x^i, y^i)\}_{i=1}^{m}$ *denote i.i.d. samples from* `LSN` *where* $m \in [cd^2, d^\alpha/c]$ *for some* $\alpha > 2$. *Then, given* $d > \Omega(\sqrt{k} \log k)$ *and initial* $w_{ij} \sim \mathcal{N}(0, \frac{1}{dk \log^4 d})$, *after* $O(1)$ *iterations, mini-batch gradient descent (over* w*) with hinge loss, constant step size, mini-batch size* $\Theta(m)$, *satisfies:*

- *Test error is at most* $1/poly(d)$
- *The learned weights of hidden units* w_{ij} *satisfy:*

$$\underbrace{|w_{1j}| = \frac{2}{\sqrt{k}}\left(1 - \frac{c}{\sqrt{\log d}}\right) + O\left(\frac{1}{\sqrt{dk}\log d}\right)}_{\textbf{\textit{Linear Coordinate}}}$$

$$\underbrace{|w_{2,j}| = O\left(\frac{1}{\sqrt{dk}\log d}\right)}_{\textbf{\textit{3-Slab Coordinate}}}, \quad \underbrace{\|w_{3:d,j}\| = O\left(\frac{1}{\sqrt{k}\log d}\right)}_{d-2 \textbf{\textit{ Noise Coordinates}}}$$

with probability $> 1 - \frac{1}{poly(d)}$. c *is a universal constant.*

Remarks: First, we see that the trained model essentially relies only on the linear coordinate w_{1j}—SGD sets the value of w_{1j} roughly $\tilde{\Omega}(\sqrt{d})$ larger than the slab coordinates w_{2j} that do not change much from their initial value. Second, the initialization we use is widely studied in the deep learning theory (Mei et al., 2018; Woodworth et al., 2020) as it better reflects practical performance of NNs (Chizat et al., 2019).

3.2. Simplicity Bias (SB) is extreme in practice

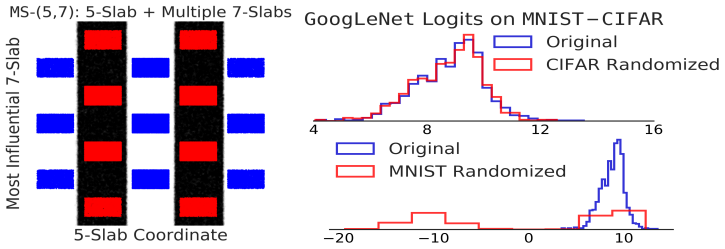
Now, we show that the SB of SGD-trained NNs is extreme. We validate the extreme nature of SB on datasets with features of varying simplicity: `LMS-5`, `MS-(5,7)`, `MNIST-CIFAR`. Recall that we $\mathtt{S}$ & $\mathtt{S}^c$ to denote the simple and (relatively) complex features in each dataset: linear & 5-slab in `LMS-5`, 5-slab & 7-slab in `MS-(5,7)`, and `MNIST` & `CIFAR` blocks in `MNIST-CIFAR`.

On `LMS-5` and `MS-(5,7)` datas, we consistently observe that fully-connected (`FCN`s) trained with SGD *exclusively* rely on the simplest feature $\mathtt{S}$ to attain standard accuracy and AUC of 1.0 but exhibit invariance to *all* complex features $\mathtt{S}^c$. Consequently, as shown in Figure 3(a), the $\mathtt{S}$-randomized AUC is 0.5 as `FCN`s exclusively rely on $\mathtt{S}$. Surprisingly, however, $\mathtt{S}^c$-randomized AUC of all models on both datasets equals 1.0; arbitrarily perturbing $\mathtt{S}^c$ coordinates has *no impact* on the class logits or predictions. The two-dimensional projections of `FCN` decision boundaries in Figure 3(c) show that our notion of simplicity extends beyond linearity and aligns with the preference of NNs: linear is simpler than 5-slab, which is simpler than 7-slab. Note that the sample size does not present any obstacles in learning complex features $\mathtt{S}^c$ to achieve *perfect* accuracy—in fact, if the simplest feature $\mathtt{S}$ is *removed* from datasets, SGD-trained models indeed rely on complex features in $\mathtt{S}^c$ to attain *perfect* accuracy.

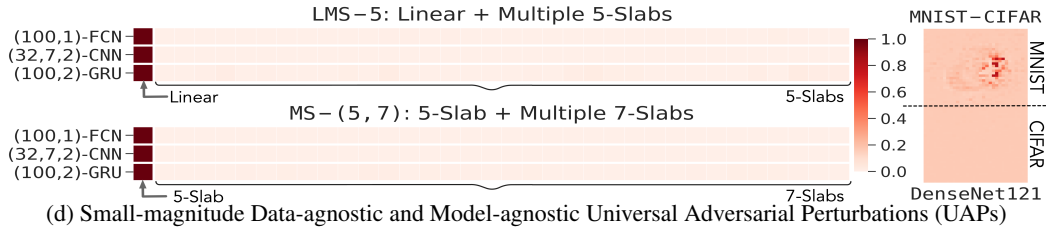
Similarly, experiments on `MNIST-CIFAR` data show that models such as GoogLeNet (Szegedy et al., 2015) exhibit extreme SB as well. Figure 3(c) shows that randomizing the simpler `MNIST` block degrades AUC to 0.5 (random chance) and randomly shuffles the logit distribution across true positives and true negatives. However, randomizing the `CIFAR` block does not alter AUC or logits, even though the `CIFAR` block alone is almost fully predictive of its label; GoogLeNet attains 95% accuracy on the corresponding CIFAR binary classification task. In the Appendix, we show that our results hold across architectures, activation func-

Dataset	Set $\mathtt{S}^c$	Size $\lvert\mathtt{S}^c\rvert$	Randomized AUC	
			Set $\mathtt{S}$	Set $\mathtt{S}^c$
LMS-5	all 5-Slabs	49	0.50 ± 0.01	1.00 ± 0.00
		249	0.50 ± 0.01	1.00 ± 0.00
MS-(5,7)	all 7-Slabs	49	0.49 ± 0.01	1.00 ± 0.00
		249	0.50 ± 0.01	1.00 ± 0.00

(a) $\{\mathtt{S}, \mathtt{S}^c\}$-Randomized AUCs of (2000,1)-FCNs



(b) (100,1)-FCN boundary (c) Logit distribution



(d) Small-magnitude Data-agnostic and Model-agnostic Universal Adversarial Perturbations (UAPs)

Figure 3: (a) $\{\mathtt{S}, \mathtt{S}^c\}$-randomized AUC vs. number of complex features, (b) FCN decision boundaries projected onto S & the most influential coordinate in S^c, and (c) Standard & $\{\mathtt{S}, \mathtt{S}^c\}$-randomized GoogLeNet logits of true positives collectively demonstrate that neural networks in practice exhibit extreme SB. Subplot (d) visualizes ℓ_2 Universal Adversarial Perturbations (UAPs) that significantly degrades model performance by almost fully utilizing its perturbation budget to attack S alone: 99.6% for linear in LMS-5, 99.9% for 5-slab in MS-(5,7) and 99.3% for MNIST pixels in MNIST-CIFAR.

tions & optimizers and are robust to ℓ_2 decay & dropout.

3.3. Extreme SB leads to Non-Robustness

Poor OOD performance: Given that NNs tend to rely on spurious features (McCoy et al., 2019; Oakden-Rayner et al., 2019), validation accuracies of state-of-the-art models on well-studied tasks provide a false sense of security; even benign distributional changes to the data can nullify model performance. This phenomenon, though counter-intuitive, can be easily explained through the lens of extreme SB. Specifically, we hypothesize that spurious features are *simple*. When combined with extreme SB, explains the outsized impact of spurious features. For example, Figure 3(c) shows that simply perturbing the simplest (and potentially spurious in practice) feature S drops the AUC of trained neural networks to 0.5, thereby nullifying model performance. However, randomizing *all* complex features $\mathtt{S}^c$ has no effect on the models; $\mathtt{S}^c$-randomized and original logits essentially overlap, even though $\mathtt{S}^c$ and S have equal predictive power. Our results also indicate that approaches (Hendrycks & Gimpel, 2016; Liang et al., 2017) that aim to detect distribution shifts based on model logits or softmax probabilities may themselves fail due to extreme SB.

Adversarial Vulnerability: Consider a $\mathcal{O}(\sqrt{d})$-margin classifier f^* that attains 100% accuracy on the LMS-5 dataset by taking an average of the linear classifier on the linear coordinate and $d-1$ piecewise-linear classifiers, one for every 5-slab coordinate. Given the large margin, f^* also attains robustness to $\mathcal{O}(\sqrt{d})$-norm ℓ_2 adversarial perturbations; attacks must perturb $\Omega(d)$ coordinates to flip model predictions. However, due to extreme SB, even large SGD-trained NNs exclusively rely on the simplest feature S and do not learn robust and large-margin classifiers such as f^*. Consequently, perturbations with norm $\mathcal{O}(1)$ suffice to nullify model performance. We validate this hypothesis in Figure 3(d), where FCNs, CNNs and GRUs (Cho et al., 2014) trained on LMS-5 & MS-(5,7) and DenseNet121 (Huang et al., 2016) trained on MNIST-CIFAR are vulnerable to small data-agnostic universal adversarial perturbations (UAPs) that target the simplest feature S. For example, Figure 3(d) shows that the ℓ_2 UAP of DenseNet121 on MNIST-CIFAR only attacks a few pixels in the simpler MNIST block and does not perturb the CIFAR block. Extreme SB also explains why data-agnostic UAPs of one model transfer well to another: the notion of simplicity is consistent across models; Figure 3(d) shows that FCNs, CNNs and GRUs trained on LMS-5 and MS-(5,7) essentially learn the same UAP. Furthermore, invariance to complex features $\mathtt{S}^c$ (e.g., CIFAR block in MNIST-CIFAR) due to extreme SB explains why "natural" (Hendrycks et al., 2019) and "semantic" perturbations (Bhattad et al., 2020) modify the true image class but not the model predictions.

4. Conclusion

We investigated Simplicity Bias (SB) in SGD-trained neural networks (NNs) using synthetic and image-based datasets that (a) incorporate a precise notion of feature simplicity, (b) are amenable to theoretical analysis and (c) capture subtleties of trained NNs in practice. We made two key contributions. First, we showed that one-hidden-layer ReLU NNs provably exhibit SB. Second, we empirically demonstrated that extreme SB in practice can result in brittle neural networks. Our results collectively motivate the need for novel algorithmic approaches aimed at avoiding the pitfalls of extreme SB, which we identify as a unifying explanation for poor OOD performance and adversarial vulnerability.

References

Arpit, D., Jastrzebski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M. S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., et al. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 233–242. JMLR. org, 2017.

Bhattad, A., Chong, M. J., Liang, K., Li, B., and Forsyth, D. Unrestricted adversarial examples via semantic manipulation. In *International Conference on Learning Representations*, 2020.

Chizat, L., Oyallon, E., and Bach, F. On lazy training in differentiable programming. In *Advances in Neural Information Processing Systems*, pp. 2933–2943, 2019.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

Gunasekar, S., Lee, J. D., Soudry, D., and Srebro, N. Implicit bias of gradient descent on linear convolutional networks. In *Advances in Neural Information Processing Systems*, pp. 9461–9471, 2018.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1321–1330. JMLR. org, 2017.

Hendrycks, D. and Gimpel, K. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., and Song, D. Natural adversarial examples. *arXiv preprint arXiv:1907.07174*, 2019.

Huang, G., Liu, Z., and Weinberger, K. Q. Densely connected convolutional networks. corr abs/1608.06993 (2016). *arXiv preprint arXiv:1608.06993*, 2016.

Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., and Madry, A. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems*, pp. 125–136, 2019.

Jo, J. and Bengio, Y. Measuring the tendency of cnns to learn surface statistical regularities. *arXiv preprint arXiv:1711.11561*, 2017.

Liang, S., Li, Y., and Srikant, R. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*, 2017.

McCoy, T., Pavlick, E., and Linzen, T. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3428–3448, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1334. URL https://www.aclweb.org/anthology/P19-1334.

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

Moosavi-Dezfooli, S.-M., Fawzi, A., Fawzi, O., and Frossard, P. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1765–1773, 2017.

Nakkiran, P., Kaplun, G., Kalimeris, D., Yang, T., Edelman, B. L., Zhang, F., and Barak, B. Sgd on neural networks learns functions of increasing complexity. *arXiv preprint arXiv:1905.11604*, 2019.

Oakden-Rayner, L., Dunnmon, J., Carneiro, G., and Ré, C. Hidden stratification causes clinically meaningful failures in machine learning for medical imaging. *arXiv preprint arXiv:1909.12475*, 2019.

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.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

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. URL https://openreview.net/forum?id=rye4g3AqFm.

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. *arXiv preprint arXiv:2002.09277*, 2020.

Supplementary Material

Simplicity Bias and the Robustness of Neural Networks

Contents

The supplementary material is organized as follows. We first discuss additional related work and provide experiment details in Appendix A and Appendix B respectively. In Appendix C, we provide additional experiments to further validate the extreme nature of Simplicity Bias (SB). Then, we provide the proof of Theorem 1 in Appendix D.

A Additional Related Work

In this section, we provide a more thorough discussion of relevant work related to the implicit bias of SGD, margin-based generalization bounds, adversarial robustness, and out-of-distribution (OOD) examples.

Implicit bias of stochastic gradient descent : Brutzkus et al. [5] shows that neural networks trained with SGD provably generalize on linearly separable data. Recent works [35, 20] also analyze the limiting direction of gradient descent on logistic regression with linearly separable and non-separable data respectively; Gunasekar et al. [15] proves similar results for linear convolutional networks. Empirical findings [28, 26] provide further evidence to suggest that neural networks trained using SGD generalize well because SGD learns models of increasing complexity over time. A few recent works have investigated the implicit bias of SGD on non-linearly separable data for linear classifiers [20] and infinite width two layer NNs [8], in both cases showing convergence to maximum margin classifiers in appropriate spaces. As discussed previously, we believe that this implicit bias of SGD (towards simplicity) can in fact be a challenge for learning robust large-margin classifiers as it is naturally biased towards simple, small-margin and feature-impoverished classifiers at the cost of feature-dense, large-margin classifiers. Our result in Theorem D.1 exhibits this phenomenon issue in a stylized setting.

Margin-based generalization bounds: Building up on the classical work of [2], recent works try to obtain tighter generalization bounds for neural networks in terms of *normalized* margin [3, 29, 11, 13]. Here, margin is defined as the difference in the probability of the true label and the largest probability of the incorrect labels. While these bounds seem to capture generalization of neural networks at a coarse level, it has been argued [27] that these approaches may be incapable of fully explaining the generalization ability of neural networks. Furthermore, it is unclear if the notion of model complexity used in these works, based on Lipschitz constant, captures generalization ability accurately. In any case, our results suggest that due to extreme simplicity bias (SB), even if a formulation captures both margin and model complexity accurately, current optimization techniques may not be able to find the optimal solution in terms of generalization *and* robustness, as they are strongly biased towards small-margin classifiers that exclusively rely on the simplest features.

Adversarial Defenses: Neural networks trained using standard procedures such as SGD are extremely vulnerable [14] to ϵ-bound adversarial attacks such as FGSM [14], PGD [25], CW [7], and Momentum [10]; Unrestricted attacks [4, 12] can significantly degrade model performance as well. Defense strategies based on heuristics such as feature squeezing [42], denoising [41], encoding [6], specialized nonlinearities [43] and distillation [30] have had limited success against stronger attacks [1]. On the other hand, standard adversarial training [25] and its variants such as [44] are fairly effective on datasets such as `MNIST`, `CIFAR-10` and `CIFAR-100`. However, on larger datasets such as `ImageNet`, these methods have limited success [34]; recent attempts [40, 34] that make adversarial training faster do not improve robustness either.

Detecting OOD Examples: Neural networks trained using standard training procedures tend to rely on low-level features and spurious correlations and hence exhibit brittleness to benign distributional changes to the data. Recent works thus aim to detect OOD examples using generative models [31], statistical tests [32], and model confidence scores [18, 23, 22]. Our experiments in Section 3 that validate extreme SB in practice also show that detectors that directly or indirectly rely on model scores to detect OOD examples may not work well as SGD-trained neural networks can exhibit complete invariance to predictive-but-complex features.

B Experiment Details

In this section, we provide additional details on the datasets, models, optimization methods and training hyperparameters used in the paper.

One-dimensional Building Blocks: We first describe the data generation process underlying each building block: linear and k-slab.

- Linear(γ, B): The linear block is parameterized by the effective margin γ and width B. The distribution first samples a label $y \in \{-1, 1\}$ uniformly at random, and then given y, x is sampled as follows: $x = y(B\gamma + (B - B\gamma) \cdot \mathrm{U}(0, 1)$, where $\mathrm{U}(0, 1)$ is the uniform distribution on $[0, 1]$.
- Slab(γ, B, k): The k-slab block is parameterized by effective margin γ, width B, and number of slabs k. We use $k \in \{3, 5, 7\}$ in our paper. The width of each slab, $w_k = {}^{2B(1-(k-1)\gamma)}/_{k}$, in the k-slab block is chosen such that the farthest points are at $-B$ and B. For example, given label $y \in \{-1, 1\}$ and random sign $z \in \{-1, 1\}$ sampled unif. at random, we can sample x from a 3-slab block as follows:

$$x = \begin{cases} z(\frac{1}{2}w_3 \cdot \mathrm{U}(0, 1)) & \text{if } y = -1 \\ z(\frac{1}{2}w_3 + 2B\gamma + w_3 \cdot \mathrm{U}(0, 1)) & \text{if } y = +1 \end{cases}$$

  For k-slab blocks with $k \in \{5, 7\}$, the probability of sampling from the two slabs (one on each side) that are farthest away from the origin are ${}^1/_4$ and ${}^1/_8$ respectively to ensure that the variance of instances in positive and negative classes, x_+ and x_-, are equal.

Datasets: We now outline the default hyperparameters for generating the synthetic datasets used in the paper, provide additional details on the `LSN` dataset, and introduce multiple variants of the `MNIST-CIFAR` dataset (i.e., with different class pairs).

- Synthetic Dataset Hyperparameters: Recall that we use four d-dimensional synthetic datasets—`LMS-k`, `L̂MS-k`, `MS-(5,7)`, and `MS-5`—wherein each coordinate corresponds to one of the building blocks described above. Unless mentioned otherwise, for all four datasets, we set the effective margin parameter $\gamma = 0.1$, width parameter $B = 1$, and noise parameter $p = 0.2$ in all blocks/coordinates. Also recall that each dataset comprises at most one "simple" feature `S` and multiple independent complex features $\mathtt{S}^c$. In our experiments, all datasets have sample sizes that are large enough for all models considered in the paper to learn complex features $\mathtt{S}^c$ and attain optimal test accuracy, even in the absence of `S`; we use sample sizes of 50000 for `LMS-5` and `MS-5` and 100000 for `L̂MS-7` and `MS-(5,7)` datasets.
- `LSN` Dataset: Recall that the `LSN` dataset (described in Section 2) is a stylized version of the `LMS-k` that is amenable to theoretical analysis. In `LSN`, conditioned on the label y, the first and second coordinates of x are *singleton* linear and 3-slab blocks: linear and 3-slab blocks have support on $\{-1, 1\}$ and $\{-1, 0, 1\}$ respectively. The remaining coordinates are standard gaussians and not predictive of the label. Each data point $(x_i, y_i) \in \Re^d \times \{-1, 1\}$ can be sampled as follows:

$$\begin{aligned} y_i = \pm 1, \;\; \text{w.p. } 1/2, \quad \varepsilon_i = \pm 1, \;\; \text{w.p. } 1/2, & \\ x_{i1} = y_i & \quad \text{(Linear coordinate)}, \\ x_{i2} = \Big(\frac{y_i + 1}{2}\Big)\varepsilon_i & \quad \text{(Slab coordinate)}, \\ x_{i,3:d} \sim \mathcal{N}(0, I_{d-2}) & \quad (d\text{–2 Noise coordinates}). \end{aligned}$$

- Additional MNIST-CIFAR Datasets: We now introduce three MNIST-CIFAR datasets, each with different MNIST and CIFAR10 classes. Recall that images in the MNIST-CIFAR datasets are concatenations of MNIST and CIFAR10 images. We introduce additional variants of the MNIST-CIFAR using different class pairs to show that our results in the paper are robust to the exact choice of pairs:

Datasets	Class -1		Class $+1$	
	MNIST	CIFAR10	MNIST	CIFAR10
MNIST-CIFAR:A	Digit 0	Automobile	Digit 1	Truck
MNIST-CIFAR:B	Digit 1	Automobile	Digit 4	Truck
MNIST-CIFAR:C	Digit 0	Airplane	Digit 1	Ship

Table 1: Three MNIST-CIFAR datasets. We use MNIST-CIFAR:A in the paper. In MNIST-CIFAR:B, we use different MNIST classes: digits 1 and 4. In MNIST-CIFAR:C, we use different CIFAR10 classes: airplane and ship. Our results in Section 3 hold on all three MNIST-CIFAR datasets.

Models: Here, we briefly describe the models (and its abbreviations) used in the paper. We use fully-connected (FCNs), convolutional (CNNs), and sequential neural networks (GRUs [9]) on synthetic datasets. Abbreviations (w, d)-FCN denotes FCN with width w and depth d, (f, k, d)-CNN denotes d-layer CNNs with f filters of size $k \times k$ in each layer with and (h, l, d)-GRU denotes d-layer d-layer GRU with input dimensionality l and hidden state dimensionality h. On MNIST-CIFAR, we train MobileNetV2 [33], GoogLeNet [37], ResNet50 [16] and DenseNet121 [19].

Training Procedures: Unless mentioned otherwise, we use the following hyperparameters for standard training and adversarial training on synthetic and MNIST-CIFAR data:

- Standard Training: On synthetic datasets, we use Stochastic Gradient Descent (SGD) with (fixed) learning rate 0.1 and batch size 256, and ℓ_2 regularization $5 \cdot 10^{-7}$. On MNIST-CIFAR datasets, we use SGD with initial learning rate 0.05 with decay factor of 0.2 every 30 epochs, momentum 0.9 and ℓ_2 regularization $5 \cdot 10^{-5}$. We do not use data augmentation. We run all models for at most 500 epochs and stop early if the training loss goes below 10^{-3}.
- Adversarial Training: We use the same SGD hyperparameters (as described above) on synthetic and MNIST-CIFAR datasets. We use Projected Gradient Descent (PGD) Adversarial Training [25] to adversarially train models. We use learning rate 0.1 and 40 iterations to generate ℓ_2 & ℓ_∞ perturbations

C Additional Results on the Extreme Nature of Simplicity Bias (SB)

Recall that Section 3 of the paper establishes the extreme nature of SB: *If all features have full predictive power, NNs rely exclusively on the simplest feature* S *and remain invariant to all complex features* $\mathtt{S}^c$—in Section 3 of the paper. Now, we further validate the extreme nature of SB across model architectures, datasets, optimizers, activation functions and regularization.

C.1 Effect of Model Architecture

In this section, we supplement our results in Section 3 of the paper by showing that extreme simplicity bias (SB) persists across several model architectures and on synthetic as well as image-based datasets.

Dataset	Set S	Set $\mathtt{S}^c$	Model	Randomized AUC	
				Set S	Set $\mathtt{S}^c$
LMS-5	Linear	5-Slabs	(100,1)-FCN	0.50	1.00
			(100,2)-FCN	0.49	1.00
			(32,7,1)-CNN	0.50	1.00
			(32,7,2)-CNN	0.50	1.00
			(100,10,1)-GRU	0.51	1.00
			(100,10,2)-GRU	0.50	1.00
MS-(5,7)	5-Slab	7-Slabs	(100,1)-FCN	0.50	1.00
			(100,1)-FCN	0.50	1.00
			(32,7,1)-CNN	0.50	1.00
			(32,7,2)-CNN	0.50	1.00
			(100,10,1)-GRU	0.50	1.00
			(100,10,2)-GRU	0.50	1.00
MNIST-CIFAR:A	MNIST block	CIFAR block	MobileNetV2	0.52	1.00
			GoogLeNet	0.51	1.00
			ResNet50	0.50	1.00
			DenseNet121	0.52	1.00

Table 2: Extreme SB across models trained on synthetic and image-based datasets show that all models exclusively rely on the simplest feature S and remain completely invariant to all complex features $\mathtt{S}^c$

In Table 2, we present {S,$\mathtt{S}^c$ }-Randomized AUCs for FCNs, CNNs and GRUs with depth {1,2} trained on LMS and MS-(5,7) datasets and state-of-the-art CNNs trained on MNIST-CIFAR:A. While the $\mathtt{S}^c$-randomized AUC equals 1.00 (perfect classification), we see that the S-randomized AUCs are approximately 0.5 for all models. This is because all models essentially only rely on the simplest feature S and remain invariant to all complex features $\mathtt{S}^c$, even though all features have equal predictive power.

C.2 Effect of MNIST-CIFAR Class Pairs

In this section, we supplement our results on MNIST-CIFAR (in Section 3) in order to show that extreme SB observed in MobileNetV2 [33], GoogLeNet [37], ResNet50 [16] and DenseNet121 [19] does not depend on the exact choice of MNIST and CIFAR10 class pairs used to construct the MNIST-CIFAR datasets. To do so, we evaluate the MNIST-randomized and CIFAR10-randomized metrics of the aforementioned models on three datasets–MNIST-CIFAR:A, MNIST-CIFAR:B, MNIST-CIFAR:C—described in Appendix B.

Model	MNIST-CIFAR:A AUCs			MNIST-CIFAR:B AUCs			MNIST-CIFAR:C AUCs		
	Standard	CIFAR10 Randomized	MNIST Randomized	Standard	CIFAR10 Randomized	MNIST Randomized	Standard	CIFAR10 Randomized	MNIST Randomized
MobileNetV2	1.00 ± 0.00	1.00 ± 0.00	0.53 ± 0.01	1.00 ± 0.00	1.00 ± 0.00	0.53 ± 0.02	1.00 ± 0.00	1.00 ± 0.00	0.50 ± 0.01
GoogLeNet	1.00 ± 0.00	1.00 ± 0.00	0.52 ± 0.02	1.00 ± 0.00	1.00 ± 0.00	0.50 ± 0.01	1.00 ± 0.00	1.00 ± 0.00	0.53 ± 0.01
ResNet50	1.00 ± 0.00	1.00 ± 0.00	0.50 ± 0.01	1.00 ± 0.00	1.00 ± 0.00	0.51 ± 0.01	1.00 ± 0.00	1.00 ± 0.00	0.50 ± 0.03
DenseNet121	1.00 ± 0.00	1.00 ± 0.00	0.53 ± 0.02	1.00 ± 0.00	1.00 ± 0.00	0.52 ± 0.01	1.00 ± 0.00	1.00 ± 0.00	0.54 ± 0.01

Table 3: (Extreme SB in three MNIST-CIFAR datasets) Standard and randomized AUCs of four state-of-the-art CNNs trained on three MNIST-CIFAR datasets. The AUC values collectively indicate that all models exclusively rely on the MNIST block.

Table 3 presents the standard, MNIST-randomized and CIFAR10-randomized AUC values of MobileNetV2, GoogLeNet, ResNet50 and DenseNet121 on three MNIST-CIFAR datasets. We observe that randomizing over the simpler MNIST block is sufficient to fully degrade the predictive power of all models; for instance, randomizing the MNIST block drops the AUC values of ResNet50 from 1.0 to 0.5 (i.e., equivalent to random classifier). However, randomizing the CIFAR10 block has no effect—standard AUC and CIFAR10-randomized AUCs equal 1.0. In stark contrast, an ideal classifier that relies on MNIST as well as CIFAR10 would attain non-trivial AUC even when the MNIST block is randomized.

C.3 Effect of Optimizers and Activation Functions

Now, we study the effect of activation function and optimizer on extreme SB. That is, can the usage of different activation functions and optimizer encourage trained neural networks to rely on complex features $\mathtt{S}^c$ in addition to the simplest feature S?

Table 4 presents the S-randomized AUCs of $(100, 2)$-FCNs with multiple activation functions—ReLU, Leaky ReLU [24], PReLU [17], and Tanh—trained on LMS-7 and MS-(5,7) datasets using multiple commonly-used optimizers: SGD, Adam [21],and RMSProp [38]. We observe that for all combinations of activations and optimizers, trained FCNs still only rely on simplest feature S; S-randomized and $\mathtt{S}^c$-randomized AUCs are approximately 0.50 and 1.0 respectively for all optimizers and activation functions. Therefore, in addition to SGD, commonly used first-order optimization methods such as Adam and RMSProp cannot jointly learn large-margin classifiers that rely on learn slab-structured features in the presence of a noisy linear structure.

To summarize, the experiment in Appendix C.2 shows that simply altering the choice of optimizer and activation function does not have any effect on extreme SB. Similar to the experiments in Section 3 of the paper, all models exclusively rely on simplest feature S and remain invariant to complex features $\mathtt{S}^c$.

Activation Function	LMS-7			MS-(5,7)		
	SGD	Adam	RMSProp	SGD	Adam	RMSProp
ReLU	0.499 ± 0.001	0.497 ± 0.003	0.502 ± 0.004	0.499 ± 0.003	0.499 ± 0.004	0.496 ± 0.004
Leaky ReLU	0.501 ± 0.001	0.497 ± 0.003	0.501 ± 0.005	0.499 ± 0.005	0.498 ± 0.002	0.498 ± 0.005
PReLU	0.500 ± 0.004	0.500 ± 0.003	0.501 ± 0.004	0.501 ± 0.004	0.496 ± 0.003	0.499 ± 0.002
Tanh	0.495 ± 0.001	0.502 ± 0.004	0.495 ± 0.004	0.498 ± 0.004	0.499 ± 0.004	0.498 ± 0.002

Table 4: (Effect of activation function and optimizers) $(100, 2)$-FCNs with multiple activation functions—ReLU, Leaky ReLU [24], PReLU [17], and Tanh—trained on LMS-5 data using common first-order optimization methods—SGD, Adam [21], and RMSProp [38]—exhibit extreme SB.

Model	Dropout	Standard AUC		S-Randomized AUC		S^c-Randomized AUC	
		$\lambda = 10^{-2}$	$\lambda = 10^{-4}$	$\lambda = 10^{-2}$	$\lambda = 10^{-4}$	$\lambda = 10^{-2}$	$\lambda = 10^{-4}$
(100,1)-FCN	0.00	1.00 ± 0.00	1.00 ± 0.00	0.50 ± 0.00	0.50 ± 0.01	1.00 ± 0.00	1.00 ± 0.00
	0.05	1.00 ± 0.00	1.00 ± 0.00	0.50 ± 0.00	0.50 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
	0.10	1.00 ± 0.00	1.00 ± 0.00	0.50 ± 0.00	0.50 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
(100,2)-FCN	0.00	1.00 ± 0.00	1.00 ± 0.00	0.50 ± 0.00	0.50 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
	0.05	1.00 ± 0.00	1.00 ± 0.00	0.50 ± 0.00	0.50 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
	0.10	1.00 ± 0.00	1.00 ± 0.00	0.50 ± 0.00	0.50 ± 0.00	1.00 ± 0.00	1.00 ± 0.00

Table 5: Dropout and ℓ_2 regularization have no effect on extreme SB of FCNs trained on LMS-7 datasets. The standard and {S,S^c }-randomized AUC values of (100,1)-FCNs and (100,2)-FCNs collectively indicate that the models still exclusively latch on to S (linear block) and remain invariant to S^c (7-slab blocks).

C.4 Effect of ℓ_2 Regularization and Dropout

In this section, we use SGD-trained FCNs trained on LMS-7 data to examine the extent to which Dropout [36] and ℓ_2 regularization alters the extreme nature of SB. Specifically, we use Dropout probability parameter $\{0.0, 0.05, 0.10\}$ and ℓ_2 regularization parameters $\{0.01, 0.001\}$ when training FCNs with width 100 and depth $\{1, 2\}$ on LMS-7 data using SGD. In Table 5, we show the standard and S^c-randomized AUCs equal 1.00 (perfect classification), whereas the S-randomized AUCs are approximately 0.5 for all models. Applying Dropout while reducing the amount of ℓ_2 regularization has negligible effect on the extreme nature of SB observed in the synthetic or image-based datasets.

D Proof of Theorem 1

In this section, we first re-introduce the data distribution and theorem. Then, we describe the proof sketch and notation, before moving on to the proof.

Linear-Slab-Noise (LSN) data: The LSN dataset is a stylized version of LMS-k that is amenable to theoretical analysis. In LSN, conditioned on the label y, the first and second coordinates of x are *singleton* linear and 3-slab blocks: linear and 3-slab blocks have support on $\{-1, 1\}$ and $\{-1, 0, 1\}$ respectively. The remaining coordinates are standard gaussians and not predictive of the label. Each data point $(x_i, y_i) \in \Re^d \times \{-1, 1\}$ from LSN can be sampled as follows:

$$\begin{aligned} y_i = \pm 1, \ \text{ w.p. } 1/2, \quad \varepsilon_i = \pm 1, \ \text{ w.p. } 1/2, & \\ x_{i1} = y_i & \text{(Linear coordinate)}, \\ x_{i2} = \Big(\frac{y_i + 1}{2}\Big)\varepsilon_i & \text{(Slab coordinate)}, \\ x_{i,3:d} \sim \mathcal{N}(0, I_{d-2}) & (d\text{–}2 \text{ Noise coordinates}). \end{aligned}$$

Then, according to Theorem 1 (restated below), NNs trained with standard mini-batch gradient descent (GD) on the LSN dataset provably learns a classifier that *exclusively* relies on the "simple" linear coordinate, thus exhibiting simplicity bias at the cost of margin.

Theorem D.1. *Let $f(x) = \sum_{j=1}^{k} v_j \cdot ReLU(\sum_{i=1}^{d} w_{i,j} x_i)$ denote a one-hidden-layer neural network with k hidden units and ReLU activations. Set $v_j = \pm 1/\sqrt{k}$ w.p. $1/2$ $\forall j \in [k]$. Let $\{(x^i, y^i)\}_{i=1}^{m}$ denote i.i.d. samples from LSN where $m \in [cd^2, d^\alpha/c]$ for some $\alpha > 2$. Then, given $d > \Omega(\sqrt{k} \log k)$ and initial $w_{ij} \sim \mathcal{N}(0, \frac{1}{dk \log^4 d})$, after $O(1)$ iterations, mini-batch gradient descent (over w) with hinge loss, constant step size, mini-batch size $\Theta(m)$, satisfies:*

- *$0 - 1$ test error is at most $\frac{1}{poly(d)}$, and*
- *The weights of learned hidden units satisfy:*

$$\begin{aligned} |w_{1j}| &= \frac{2}{\sqrt{k}} \left(1 - \frac{c}{\sqrt{\log d}}\right) + O\left(\frac{1}{\sqrt{dk} \log d}\right) && \textit{(Linear coordinate)} \\ |w_{2,j}| &= O\left(\frac{1}{\sqrt{dk} \log d}\right) && \textit{(Slab coordinate)} \\ \|w_{3:d,j}\| &= O\left(\frac{1}{\sqrt{k} \log d}\right) && \textit{(Noise coordinates)} \end{aligned}$$

with probability greater than $1 - \frac{1}{poly(d)}$. Note that c is a universal constant.

Proof Sketch Since the number of iterations $t = O(1)$, we partition the dataset into t minibatches each of size $n := m/t$ samples. This means that each iteration uses a fresh batch of n samples and the t iterations together form a single pass over the data. The overall outline of the proof is as follows. If the step size is η, then for $t \lesssim \frac{4}{\eta}$ iterations, with probability $\geq 1 - \frac{1}{\text{poly}(d)}$,

- Lemma D.3 shows that the hinge loss is "active" (i.e., $yf(x) < 1$) for all data points in a given batch.

- Under this condition, we derive closed-form expressions for *population* gradients in Lemmas D.5, D.6 and D.7.
- Lemma D.2 uses the above lemmas to establish precise estimates of the linear, slab and noise coordinates for all iterations until t.

The appendix is organized as follows. Appendix D.1 presents the main lemmas that will directly lead to Theorem D.1. Appendix D.2 derives closed form expressions for population gradients and Appendix D.3 presents auxiliary lemmas that are useful in the main proofs.

Notation Recall that $f(x) = \sum_{j=1}^{k} v_j \cdot \text{ReLU}(\sum_{i=1}^{d} w_{ij}x_i) = v^T\text{ReLU}(W^Tx)$ where $W \in \mathbb{R}^{d\times k}$ and $v \in \mathbb{R}^k$. Note that $w_i = [w_{1i}\, w_{2i}, \cdots w_{di}]^T$ is the i^{th} column in W. Let $\bar{w}_i$ and $\bar{x}_j$ denote the $w_{3:d,i}$ and $x_{3:d,j}$ respectively. Also, let $\mathcal{S}_n = \{(x_i, y_i)\}_{i=1}^n$ denote a set of n i.i.d. points randomly sampled from LSN. For simplicity, we also assume $|\{i : v_i = 1/\sqrt{k}\}| = |\{i : v_i = -1/\sqrt{k}\}| = k/2$. We can now define the loss function as $\mathcal{L}_f(\mathcal{S}_n) = 1/n \sum_i^n \ell(x_i, y_i)$, where $\ell(x, y) = \max(0, 1 - yf(x))$ denotes the hinge loss. For notational simplicity, we use $X = \mu \pm \delta$ and $|X - \mu| \le \delta$ interchangeably. Also let φ and ϕ denote the probability density function and cumulative distribution function of standard normal distribution.

Proof of Theorem D.1. The proof directly follows from Lemma D.2 and Lemma D.3. In Lemma D.2, we show that the weights in the linear coordinate are $\Omega(\sqrt{d})$ larger than the weights in the slab and noise coordinates. Applying Lemma D.2 at $\hat{t} = \lfloor \frac{4}{\eta}(1 - \frac{c_n}{\sqrt{\log d}}) \rfloor$ gives the following result:

$$w_{1i}^{(\hat{t})} \overset{(a)}{=} \frac{2}{\sqrt{k}}(1 - \frac{c_n}{\sqrt{\log d}}) + O(\frac{1}{\sqrt{dk}\log d}) \text{ and } |w_{2i}^{(\hat{t})}| \overset{(a)}{=} O(\frac{1}{\sqrt{dk}\log d}) \text{ and } ||\bar{w}_i^{(\hat{t})}| \overset{(a)}{=} O(\frac{1}{\sqrt{k}\log d})$$

where (a) is due to $c_0(1 + \hat{c})^t \le c_0 e^{\hat{c}t} \le c_0 e^1 = O(1)$.

The $0 - 1$ error of the function f at timestep $\hat{t}$ is small as well, because we can directly use Lemma D.3 to get $\Pr(yf(x) < 0) = 2/c^3d^6$. Therefore, the $0 - 1$ error is at most $\frac{2}{c^3d^6} = O(\frac{1}{d^6})$. ■

D.1 Proof by Induction

In this section, we use proof by induction to show that for the first $t = O(1/\eta)$ steps, (1) the hinge loss is "active" for all data points (Lemma D.3) and (2) hidden layer weights in the linear coordinate are $\Omega(\sqrt{d})$ larger than the hidden layer weights in the slab and noise coordinates (Lemma D.2).

Lemma D.2. *Let $|\mathcal{S}_n| \in [cd^2, d^\alpha/c]$ and initialization $w_{ij} \sim \mathcal{N}(0, 1/dk\log^2 d)$. Also let $\hat{c} = \eta/4$, $c_0 = 2$ and $c_n = 5\sqrt{\alpha}c_0(1 + \hat{c})^t$. Then, for all $t \le \frac{4}{\eta}(1 - c_n/\sqrt{\log d})$, $d \ge \exp((8c_n/\eta)^2)$, $\sqrt{d/\log^3(d)} > 24\sqrt{k}/c_0c$ and $i \in [k]$, w.p. greater than $1 - O(\frac{1}{d^2})$, we have:*

$$y_i f(x_i) \le 1 \,\forall (x_i, y_i) \in \mathcal{S}_n \tag{1}$$

$$w_{1i}^{(t)} = \frac{t\eta v_i}{2} \pm \frac{c_0(1 + \hat{c})^t}{\sqrt{dk}\log d} \tag{2}$$

$$|w_{2i}^{(t)}| \le \frac{c_0(1 + \hat{c})^t}{\sqrt{dk}\log d} \tag{3}$$

$$||\bar{w}^{(t)}||_2 \le \frac{c_0(1 + \hat{c})^t}{\sqrt{k}\log d} \tag{4}$$

Proof. First, we prove that equations (2), (3) & (4) hold at initialization (i.e., $t = 0$) with high probability. Using D.8 and D.13:

$$\max_{i\in\{1,2\}} \max_{j\le k} |w_{ij}| \le \frac{2}{\sqrt{dk}\log d} \quad \text{and} \quad \max_{i\le k} ||\bar{w}_i|| \le \frac{2}{\sqrt{k}\log d} \quad \text{w.p. } 1 - \frac{2}{d^4}$$

Therefore, $w_{1i} = \frac{(0)\eta v_i}{2} \pm \frac{c_0(1+\hat{c})^0}{\sqrt{dk}\log d}$ and $|w_{2i}| \le \frac{c_0(1+\hat{c})^0}{\sqrt{dk}\log d}$ and $||\bar{w}_i|| \le \frac{c_0(1+\hat{c})^0}{\sqrt{k}\log d}$. Since equations (2), (3) & (4) hold at $t = 0$, we can use Lemma D.3 to show that the hinge loss is "active" with high probability:

$$y_i f(x_i) = \pm\frac{c_n}{\sqrt{\log d}} < 1 \quad \text{when } d \ge \exp(c_n^2)$$

Now, we assume that the inductive hypothesis—equations (1), (2), (3) and (4)—is true after every timestep τ where $\tau \in \{0, \cdots, t\}$.

We now prove that the inductive hypothesis is true at timestep $t+1$, after applying gradient descent using the $(t+1)^{th}$ batch. Since z(1) holds at timestep t, we can use the closed-form expression of the gradient along the linear coordinate (lemma D.5) to prove that equation (2) holds at timestep $t+1$ as well:

$$\begin{aligned}
w_{1i}^{(t+1)} &= w_{1i}^{(t)} + \frac{\eta v_i}{4}\Big[2 + \phi\Big(\frac{w_{1i}+w_{2i}}{||\bar{w}_i||}\Big) + \phi\Big(\frac{w_{1i}-w_{2i}}{||\bar{w}_i||}\Big) - 2\phi\Big(\frac{w_{1i}}{||\bar{w}_i||}\Big)\Big] \pm \frac{5\eta v_i}{d}\sqrt{\frac{\log(cd^2)}{c}} \\
&= w_{1i}^{(t)} + \frac{\eta v_i}{2} + \frac{\eta v_i}{2}|w_{2i}^{(t)}| \cdot \max_{|\delta|\le|w_{2i}^{(t)}|} \frac{1}{||\bar{w}^t||}\varphi\Big(\frac{w_{1i}^{(t)}+\delta}{||\bar{w}_i^{(t)}||}\Big) \pm \frac{5\eta v_i}{d}\sqrt{\frac{\log(cd^2)}{c}} \\
&\overset{(a)}{=} \frac{t\eta v_i}{2} \pm \frac{c_0(1+\hat{c})^t}{\sqrt{dk}\log d} + \frac{\eta v_i}{2} + \frac{\eta v_i}{2}\frac{c_0(1+\hat{c})^t}{\sqrt{dk}\log d} \pm \frac{5\eta v_i}{d}\sqrt{\frac{\log(cd^2)}{c}} \\
&\overset{(b)}{=} \frac{(t+1)\eta v_i}{2} \pm \frac{c_0(1+\hat{c})^t}{\sqrt{dk}\log d} \pm \eta v_i\frac{c_0(1+\hat{c})^t}{\sqrt{dk}\log d} = \frac{(t+1)\eta v_i}{2} \pm \frac{c_0(1+\hat{c})^t(1+\eta v_i)}{\sqrt{dk}\log d} \\
&\overset{(c)}{=} \frac{(t+1)\eta v_i}{2} \pm \frac{c_0(1+\hat{c})^{t+1}}{\sqrt{dk}\log d}
\end{aligned}$$

where (a) is via equation (12) in Lemma D.4, (b) is because $d/\log^3(d) \ge 20/c_0 e^1\sqrt{c}$ and (c) is due to $\eta v_i \le \hat{c}$.

Similarly, since equation (1) holds at timestep t (via the inductive hypothesis), we can use the closed-form expression of the gradient along the slab coordinate (lemma D.6) to show that the weights in the slab (i.e., second) coordinate are small (equation (3)) at timestep $t+1$ as well:

$$\begin{aligned}
w_{2i}^{(t+1)} &= w_{2i}^{(t)} + \frac{\eta v_i}{4}\Big[\phi\Big(\frac{w_{1i}+w_{2i}}{||\bar{w}_i||}\Big) - \phi\Big(\frac{w_{1i}-w_{2i}}{||\bar{w}_i||}\Big)\Big] \pm \frac{5\eta v_i}{d}\sqrt{\frac{\log(cd^2)}{c}} \\
&= w_{2i}^{(t)} + \frac{\eta v_i}{2}|w_{2i}^{(t)}| \cdot \max_{|\delta|\le|w_{2i}^{(t)}|} \frac{1}{||\bar{w}^t||}\varphi\Big(\frac{w_{1i}^{(t)}+\delta}{||\bar{w}_i^{(t)}||}\Big) \pm \frac{5\eta v_i}{d}\sqrt{\frac{\log(cd^2)}{c}} \\
&\overset{(a)}{=} \pm\frac{c_0(1+\hat{c})^t}{\sqrt{dk}\log d} \pm \frac{\eta v_i}{2}\frac{c_0(1+\hat{c})^t}{\sqrt{dk}\log d} \pm \frac{\eta v_i}{2}\frac{c_0(1+\hat{c})^t}{\sqrt{dk}\log d} \le \pm\frac{c_0(1+\hat{c})^{t+1}}{\sqrt{dk}\log d}
\end{aligned}$$

where (a) is due to equations (11) in Lemma D.4, (3) and $d/\log^3(d) \ge 20/c_0 e^1\sqrt{c}$.

Finally, we can use the closed-form expression of the gradient along the noise coordinate (lemma D.7) to prove that the norm of the gradient along the noise coordinates (i.e., coordinates 3 to d) is small (equation (4)) at timestep $t+1$:

$$\bar{w}_i^{(t+1)} = \bar{w}_i^{(t+1)} + \underbrace{\frac{\eta v_i}{4}\Big[\varphi\Big(\frac{w_{1i}+w_{2i}}{||\bar{w}_i||}\Big) + \varphi\Big(\frac{w_{1i}-w_{2i}}{||\bar{w}_i||}\Big) - 2\varphi\Big(\frac{w_{1i}}{||\bar{w}_i||}\Big)\Big]\frac{\bar{w}_i^{(t)}}{||\bar{w}_i||}}_{\bar{g}_1} \pm \underbrace{\frac{3\eta|v_i|\log(\sqrt{c}d)}{\sqrt{cd}}\frac{\bar{w}_i}{||\bar{w}_i||} \pm \frac{6\eta|v_i|}{\sqrt{cd}}u_i^{\perp}}_{\bar{g}_2}$$

We first show that the the first part of the noise gradient, $\bar{\mathcal{G}}_1$, is at most $\eta v_i/2$:

$$\frac{\eta v_i}{4}\Big[\varphi\Big(\frac{w_{1i}+w_{2i}}{||\bar{w}_i||}\Big)+\varphi\Big(\frac{w_{1i}-w_{2i}}{||\bar{w}_i||}\Big)-2\varphi\Big(\frac{w_{1i}}{||\bar{w}_i||}\Big)\Big]$$
$$=\frac{\eta v_i}{4}\underbrace{\frac{1}{||\bar{w}_i^{(t)}||}\varphi\Big(\frac{w_{1i}^{(t)}}{||\bar{w}_i^{(t)}||}\Big)}_{\leq 1 \text{(see eq. 13 in Lemma D.4)}}\underbrace{\Big[2-\varphi\Big(\frac{w_{2i}^{(t)}}{||\bar{w}_i^{(t)}||}\Big)\Big(\exp(\frac{w_{1i}^{(t)}w_{2i}^{(t)}}{||\bar{w}_i^{(t)}||^2})+\exp(\frac{-w_{1i}^{(t)}w_{2i}^{(t)}}{||\bar{w}_i^{(t)}||^2})\Big)\Big]}_{\leq 2}\leq\frac{\eta v_i}{2}$$

Next, we show that the ℓ_2 norm of the second part of the noise gradient, $||\bar{\mathcal{G}}_2||$, is $O(1/\sqrt{d})$:

$$||B||\leq 3\eta|v_i|\frac{\log(\sqrt{c}d)}{\sqrt{c}d}+\frac{6\eta|v_i|}{\sqrt{cd}}\leq\frac{12\eta|v_i|}{\sqrt{cd}}$$

Now, we can use the upper bounds on $\mathcal{G}_1$ and $\mathcal{G}_2$ to show that the ℓ_2 norm of the gradient along the noise gradients is small as well:

$$||\bar{w}_i^{(t+1)}||\leq||\bar{w}_i^{(t)}||+\frac{\eta v_i}{2}||\bar{w}_i^{(t)}||+\frac{12\eta|v_i|}{\sqrt{cd}}\overset{(a)}{\leq}||\bar{w}_i^{(t)}||+\frac{\eta v_i}{2}||\bar{w}_i^{(t)}||+\frac{\eta v_i}{2}||\bar{w}_i^{(t)}||$$
$$\overset{(b)}{\leq}\frac{c_0(1+\hat{c})^t(1+\eta v_i)}{\sqrt{k}\log d}\overset{(c)}{\leq}\frac{c_0(1+\hat{c})^{t+1}}{\sqrt{k}\log d}$$

where (a) is because $d/\log d\geq(24\sqrt{k}/c_0c)^2$, (b) is due to equation (4) and (c) is because $\eta v_i\leq\hat{c}$. ■

Since equations (2), (3) & (4) hold at timestep t (from Lemma D.2), we can show that the hinge loss is positive (i.e., $yf(x)<1$) for all data points with high probability as well.

Lemma D.3. *Let $\mathcal{S}_n$ denote a set of $n\in[cd^2,d^\alpha/c]$ i.i.d. samples from LSN, where $\alpha>2$ and $c>1$. Suppose equations (2), (3) & (4) hold at timestep t. Also let $d\geq\exp((\frac{8c_n}{\eta})^2)$ where $c_n=5\sqrt{\alpha}c_0(1+\hat{c})^t$. Then, w.p. greater than $1-\frac{2}{c^3d^6}$, we have:*

$$y_if(x_i)=\frac{t\eta}{4}\pm\frac{c_n}{\sqrt{\log d}}=(t\pm 1/2)\frac{\eta}{4}\quad\forall(x_i,y_i)\in\mathcal{S}_n \tag{5}$$

Proof. We use equations (2), (3) & (4) to obtain simplify the dot product between $w_i^{(t)}$ & x_j and the indicator $\mathbb{1}\Big\{w_i^{(t)}\cdot x_j\geq 0\Big\}$. First, we show that the dot product between $w_i^{(t)}$ and x_j is in the band $\frac{t\eta v_iy_j}{2}\pm\frac{c_n}{\sqrt{k\log d}}$ with high probability:

$$w_i^{(t)}\cdot x_j=w_{1i}^{(t)}y_j+w_{2i}^{(t)}\frac{y_j+1}{2}\varepsilon_j+\bar{w}_i^{(t)}\cdot\bar{x}_j\overset{(a)}{=}w_{1i}^{(t)}y_j+w_{2i}^{(t)}\frac{y_j+1}{2}\varepsilon_j+||\bar{w}_i^{(t)}||Z_j$$
$$\overset{(b)}{=}w_{1i}^{(t)}y_j+w_{2i}^{(t)}\frac{y_j+1}{2}\varepsilon_j\pm||\bar{w}_i^{(t)}||\sqrt{8\alpha\log d}\qquad\text{w.p. } 1-\frac{2}{d^6}$$
$$=\frac{t\eta v_iy_j}{2}\pm\frac{c_0(1+\hat{c})^t}{\sqrt{dk}\log d}(y_j+\frac{y_j+1}{2}\varepsilon_j)\pm\frac{c_0(1+\hat{c})^t}{\sqrt{k}\log d}\sqrt{8\alpha\log d}\qquad\text{via eq. 2, 3, 4}$$
$$\overset{(c)}{=}\frac{t\eta v_iy_j}{2}\pm\frac{2c_0(1+\hat{c})^t}{\sqrt{dk}\log d}\pm\frac{3\sqrt{\alpha}c_0(1+\hat{c})^t}{\sqrt{k}\log d}=\frac{t\eta v_iy_j}{2}\pm\frac{c_n}{\sqrt{k}\log d} \tag{6}$$
$$=(ty_j\pm 1/4)\frac{\eta v_i}{2}\quad\text{when } d\geq\exp((\frac{8c_n}{\eta})^2) \tag{7}$$

where (a) is because $\bar{w}_i^{(t)} \cdot \bar{x}_j = ||\bar{w}_i^{(t)}||\mathcal{N}(0,1)$, (b) is via lemma D.8 & $c > 1$, and (c) is because $(y_j + \frac{y_j+1}{2}\varepsilon_j) < 2$. Next, when $d \geq \exp((\frac{8c_n}{\eta})^2)$, we can simplify $\mathbb{1}\Big\{w_i^{(t)} \cdot x_j \geq 0\Big\}$ as follows:

$$\mathbb{1}\Big\{w_i^{(t)} \cdot x_j \geq 0\Big\} \overset{eq.7}{\leq} \mathbb{1}\Big\{(ty_j \pm 1/4)\frac{\eta v_i}{2} \geq 0\Big\} \leq \begin{cases} 1, & \text{if } t = 0 \\ 1, & \text{if } t > 0 \text{ and } y_j v_i \geq 0 \\ 0, & \text{if } t > 0 \text{ and } y_j v_i < 0 \end{cases} \leq \mathbb{1}\{t = 0 \vee yv_i \geq 0\} \tag{8}$$

We can now use equations (6) & (8) to show that $y_j f^{(t)}(x_j)$ is in the band $t\eta/4 \pm O(1/\sqrt{\log d})$ with high probability:

$$\begin{aligned} y_j f^{(t)}(x_j) &= \sum_{i=1}^{k} y_j v_i \cdot \text{ReLU}(w_i \cdot x_j) = \sum_{i=1}^{k} v_i \mathbb{1}\{t = 0 \vee yv_i \geq 0\}\big(\frac{t\eta v_i}{2} \pm \frac{c_n}{\sqrt{k \log d}}\big) \\ &= \sum_{i=1}^{k} \mathbb{1}\{t = 0 \vee yv_i \geq 0\}\big(\frac{t\eta}{2k} \pm \frac{c_n}{k\sqrt{\log d}}\big) \overset{(a)}{=} \begin{cases} \pm k\frac{c_n}{k\sqrt{\log d}}, & \text{if } t = 0 \\ \frac{k}{2}\big(\frac{t\eta}{2k} \pm \frac{c_n}{k\sqrt{\log d}}\big), & \text{if } t > 0 \end{cases} \\ &= \frac{t\eta}{4} \pm \frac{c_n}{\sqrt{\log d}} \overset{(b)}{=} (t \pm 1/2)\frac{\eta}{4} \end{aligned} \tag{9}$$

where (a) is due to $|\{v_i \mid v_i > 0\}| = |\{v_i \mid v_i < 0\}| = k/2$ and (b) follows from $c_n/\sqrt{\log d} \leq \eta/8$ when $d \geq \exp((8c_n/\eta)^2)$ ■

Lemma D.4. *If equations (2), (3) & (4) hold at timestep t, $d > \exp((\frac{4c_0 e^1}{\eta})^2)$ and $d/\log d > \sqrt{k}$, we have:*

$$\max_{|\delta| \leq |w_{2i}^{(t)}|} \frac{1}{||\bar{w}^t||}\varphi\Big(\frac{w_{1i}^{(t)} + \delta}{||\bar{w}_i^{(t)}||}\Big) \leq 1 \tag{10}$$

$$\Big|\phi\Big(\frac{w_{1i}^{(t)} + w_{2i}^{(t)}}{||\bar{w}^{(t)}||}\Big) - \phi\Big(\frac{w_{1i}^{(t)} - w_{2i}^{(t)}}{||\bar{w}^{(t)}||}\Big)\Big| \leq \frac{2c_0(1+\hat{c})^t}{\sqrt{dk}\log d} \tag{11}$$

$$\Big|\phi\Big(\frac{w_{1i}^{(t)} + w_{2i}^{(t)}}{||\bar{w}^{(t)}||}\Big) - \phi\Big(\frac{w_{1i}^{(t)}}{||\bar{w}^{(t)}||}\Big)\Big| \leq \frac{c_0(1+\hat{c})^t}{\sqrt{dk}\log d} \tag{12}$$

$$\frac{1}{||\bar{w}^t||}\varphi\Big(\frac{w_{1i}^{(t)}}{||\bar{w}_i^{(t)}||}\Big) \leq \frac{c_0(1+\hat{c})^t}{\sqrt{dk}\log d} \tag{13}$$

Proof. Let $g_z(x) = \frac{1}{x}\varphi(\frac{z}{x})$ and $h(x) = \max_{|\delta| \leq |w_{2i}^{(t)}|} \frac{1}{x}\varphi(\frac{w_{1i}^{(t)}+\delta}{x}) = \max_{|\delta| \leq |w_{2i}^{(t)}|} g_{w_{1i}^t + \delta}(x)$. To prove Equation (10), we show that an upper bound on $||w^{(t)}||$ is less than a lower bound on $\arg\max_x h(x)$, which subsequently implies that $h(||w^{(t)}||) < \max_x h(x)$ because h is an increasing function for all $|x| \leq \arg\max_x h(x)$.

First, we find the maximizer x^* of $h(x)$ as follows:

$$\max_x h(x) = \max_x \max_{|\delta| \leq |w_{2i}^{(t)}|} g_{w_{1i}^{(t)}+\delta}(x) \overset{(a)}{=} \max_{|\delta| \leq |w_{2i}^{(t)}|} \frac{e^{-1}}{|w_{1i}^{(t)} + \delta|} \qquad \text{when } x^* = |w_{1i}^{(t)} + \delta|$$

where (a) follows from lemma D.12. Next, we lower bound the maximizer x^* of $h(x)$:

$$x^* = |w_{1i}^{(t)} + \delta| \geq |w_{1i}^{(t)} + w_{2i}^{(t)}| \geq \Big||w_{1i}^{(t)}| - |w_{2i}^{(t)}|\Big| \overset{(a)}{\geq} \Big||\frac{t\eta v_i}{2} \pm \frac{c_0(1+\hat{c})^t}{\sqrt{dk}\log d}| - |\frac{c_0(1+\hat{c})^t}{\sqrt{dk}\log d}|\Big|$$

$$\overset{(b)}{\geq} \Big|\Big|\frac{t\eta v_i}{2} \pm \frac{\eta v_i}{8}\Big| - \Big|\frac{\eta v_i}{8}\Big|\Big| \geq \Big|t - 1/2\Big|\frac{\eta v_i}{2} \geq \frac{\eta v_i}{4}$$

where (a) follows from the weights in the linear and slab coordinate at timestep t (equations (2) & (3)) and (b) is because $\frac{c_0(1+\hat{c})^t}{\sqrt{dk}\log d} \leq \frac{\eta v_i}{8}$ when $\sqrt{d} \geq \frac{8c_0e^1}{\eta}$. Therefore, $\arg\max_x h(x) \geq \eta v_i/4$. We can use the upper bound on the ℓ_2 norm of the gradient along the noise coordinates (equation (4)) and $d \geq \exp(\frac{4c_0e^1}{\eta})$ to show that $||\bar{w}_i^{(t)}||$ is less than x^*:

$$||\bar{w}_i^{(t)}|| \leq \frac{c_0(1+\hat{c})^t}{\sqrt{dk}\log d} \leq \frac{\eta v_i}{4} \leq \arg\max_x h(x)$$

From lemma D.12, we know that $h(x)$ is an increasing function for all $|x| < x^*$. This implies that $h(||w_i^{(t)}||) \leq h(\frac{c_0(1+\hat{c})^t}{\sqrt{dk}\log d}) \leq h(\frac{\eta v_i}{4}) \leq h(x^*)$. Therefore, when $d \geq \exp((\frac{4c_0e^1}{\eta})^2)$ and $d/\log d \geq \sqrt{k}$, we obtain the desired result as follows:

$$\max_{|\delta|\leq|w_{2i}^{(t)}|} \frac{1}{||\bar{w}^t||}\varphi\Big(\frac{w_{1i}^{(t)}+\delta}{||\bar{w}_i^{(t)}||}\Big) = h(||w_i^{(t)}||) \leq h\Big(\frac{c_0e^1}{\sqrt{dk}\log d}\Big) \leq \frac{\sqrt{k}\log d}{c_0e^1}\frac{1}{d^{(\frac{\eta}{4c_0e^1})^2\log d}} \leq 1$$

Now, we can prove equations (11), (12) and (13) using equation (10) as follows:

$$\Big|\phi\Big(\frac{w_{1i}^{(t)}+w_{2i}^{(t)}}{||\bar{w}^{(t)}||}\Big) - \phi\Big(\frac{w_{1i}^{(t)}-w_{2i}^{(t)}}{||\bar{w}^{(t)}||}\Big)\Big| \leq 2|w_{2i}^{(t)}| \cdot \max_{|\delta|\leq|w_{2i}^{(t)}|}\frac{1}{||\bar{w}^t||}\varphi\Big(\frac{w_{1i}^{(t)}+\delta}{||\bar{w}_i^{(t)}||}\Big) \overset{(a)}{\leq} \frac{2c_0(1+\hat{c})^t}{\sqrt{dk}\log d} \tag{14}$$

$$\Big|\phi\Big(\frac{w_{1i}^{(t)}+w_{2i}^{(t)}}{||\bar{w}^{(t)}||}\Big) - \phi\Big(\frac{w_{1i}^{(t)}}{||\bar{w}^{(t)}||}\Big)\Big| \leq |w_{2i}^{(t)}| \cdot \max_{|\delta|\leq|w_{2i}^{(t)}|}\frac{1}{||\bar{w}^t||}\varphi\Big(\frac{w_{1i}^{(t)}+\delta}{||\bar{w}_i^{(t)}||}\Big) \overset{(a)}{\leq} \frac{c_0(1+\hat{c})^t}{\sqrt{dk}\log d} \tag{15}$$

$$\frac{1}{||\bar{w}^t||}\varphi\Big(\frac{w_{1i}^{(t)}}{||\bar{w}_i^{(t)}||}\Big) \leq \max_{|\delta|\leq|w_{2i}^{(t)}|}\frac{1}{||\bar{w}^t||}\varphi\Big(\frac{w_{1i}^{(t)}+\delta}{||\bar{w}_i^{(t)}||}\Big) \overset{(a)}{\leq} \frac{c_0(1+\hat{c})^t}{\sqrt{dk}\log d} \tag{16}$$

where (a) is due to equation (4). ■

D.2 Closed-form Gradient Expressions

In this section, we provide closed-form expressions for gradients along the linear, slab and noise coordinates: $\nabla_{w_{1i}}\mathcal{L}_f(\mathcal{S}_n)$, $\nabla_{w_{2i}}\mathcal{L}_f(\mathcal{S}_n)$ and $\nabla_{\bar{w}_i}\mathcal{L}_f(\mathcal{S}_n)$. First, we provide a closed-form expression for the gradient along the linear coordinate:

Lemma D.5. *If $n > cd^2$ and $y_i f(x_i) < 1\,\forall(x_i, y_i) \in \mathcal{S}_n$, then w.p. greater than $1 - \frac{3}{n}$:*

$$\nabla_{w_{1i}}\mathcal{L}_f(\mathcal{S}_n) = -\frac{v_i}{4}\Big[2 + \phi\Big(\frac{w_{1i}+w_{2i}}{||\bar{w}_i||}\Big) + \phi\Big(\frac{w_{1i}-w_{2i}}{||\bar{w}_i||}\Big) - 2\phi\Big(\frac{w_{1i}}{||\bar{w}_i||}\Big)\Big] \pm \frac{5v_i}{d}\sqrt{\frac{\log(cd^2)}{c}}$$

Proof.

$$\nabla_{w_{1i}}\mathcal{L}_f(\mathcal{S}_n) = -\frac{v_i}{n}\sum_{j=1}^{n}\mathbb{1}\{y_j f(x_j) \leq 1\}\mathbb{1}\{w_i^T x_j \geq 0\}y_j x_{1j}$$

$$\overset{(a)}{=} -\frac{v_i}{n}\sum_{j=1}^{n}\mathbb{1}\{\bar{w}_i^T\bar{x}_j \geq -w_{i1}y_j - w_{i2}\mathbb{1}\{y_j = 1\}\varepsilon_j\}$$

$$\overset{(b)}{=} -\frac{v_i}{n}\sum_{j=1}^{n}\mathbb{1}\left\{Z_j \geq \frac{-w_{1i}y - w_{2i}\mathbb{1}\{y_j=1\}\varepsilon_j}{||\bar{w}_i||}\right\} \qquad \text{where } Z_j \sim \mathcal{N}(0,1)$$

$$= -v_i \sum_{l}^{\{0,1,-1\}} \frac{1}{n}\sum_{i=1}^{n}\mathbb{1}\{x_{2j}=l\}\mathbb{1}\left\{Z_j \geq \frac{-w_{1i}(2l^2-1)-w_{2i}l}{||\bar{w}_i||}\right\}$$

$$= -v_i \sum_{l}^{\{0,1,-1\}} \Big(\mathbb{P}(x_{2j}=l)\phi\Big(\frac{w_{1i}(2l^2-1)+w_{2i}l}{||\bar{w}_i||}\Big) \pm \sqrt{\frac{\log n}{n}}\Big) \qquad \text{via lemma D.10}$$

$$= -\frac{v_i}{4}[\phi(\frac{w_{1i}+w_{2i}}{||\bar{w}_i||}) + \phi(\frac{w_{1i}-w_{2i}}{||\bar{w}_i||}) + 2\phi(\frac{-w_{1i}}{||\bar{w}_i||})] \pm \frac{5v_i}{d}\sqrt{\frac{\log(cd^2)}{c}} \qquad n > cd^2$$

$$= -\frac{v_i}{4}\Big[2 + \phi\Big(\frac{w_{1i}+w_{2i}}{||\bar{w}_i||}\Big) + \phi\Big(\frac{w_{1i}-w_{2i}}{||\bar{w}_i||}\Big) - 2\phi\Big(\frac{w_{1i}}{||\bar{w}_i||}\Big)\Big] \pm \frac{5v_i}{d}\sqrt{\frac{\log(cd^2)}{c}} \qquad \text{w.p. } 1-\frac{3}{n}$$

where (a) is due to $y_i x_{i1} = y_i^2 = 1$ & $\mathbb{1}\{y_j f(x_j) \leq 1\} = 1$ and (b) is due to $\mathbb{1}\{\bar{w}_i^T \bar{x}_j \geq k\} = \mathbb{1}\{||\bar{w}_i|| Z_j \geq k\}$. ■

Similarly, we provide a closed-form expression for the gradient along the slab coordinate:

Lemma D.6. *If $n > cd^2$ and $y_i f(x_i) < 1\, \forall (x_i, y_i) \in \mathcal{S}_n$, then w.p. greater than $1 - \frac{3}{n}$:*

$$\nabla_{w_{2i}} \mathcal{L}_f(\mathcal{S}_n) = -\frac{v_i}{4}\Big[\phi\Big(\frac{w_{1i}+w_{2i}}{||\bar{w}_i||}\Big) - \phi\Big(\frac{w_{1i}-w_{2i}}{||\bar{w}_i||}\Big)\Big] \pm \frac{5v_i}{d}\sqrt{\frac{\log(cd^2)}{c}}$$

Proof.

$$\nabla_{w_{2i}} \mathcal{L}_f(\mathcal{S}_n) = -\frac{v_i}{n}\sum_{j=1}^{n}\mathbb{1}\{y_j f(x_j) \leq 1\}\mathbb{1}\big\{w_i^T x_j \geq 0\big\} y_j x_{2j}$$

$$\overset{(a)}{=} -\frac{v_i}{n}\sum_{j=1}^{n}\mathbb{1}\big\{\bar{w}_i^T \bar{x}_j \geq -w_{1i}y_j - w_{2i}\mathbb{1}\{y_j=1\}\varepsilon_j\big\}\mathbb{1}\{y_j=1\}\varepsilon_j$$

$$\overset{(b)}{=} v_i \sum_{l}^{\{-1,1\}} \frac{1}{n}\sum_{j=1}^{n}(-1)^{\mathbb{1}\{\varepsilon_j=l\}}\mathbb{1}\left\{Z_j \geq \frac{-w_{1i}-w_{2i}l}{||\bar{w}_i||}\right\}$$

$$= -v_i \sum_{l}^{\{1,-1\}} \Big(\mathbb{P}(x_{2j}=l)\phi\Big(\frac{w_{1i}+w_{2i}l}{||\bar{w}_i||}\Big) \pm \sqrt{\frac{\log n}{n}}\Big) \qquad \text{via lemma D.10}$$

$$= -\frac{v_i}{4}[\phi(\frac{w_{1i}+w_{2i}}{||\bar{w}_i||}) - \phi(\frac{w_{1i}-w_{2i}}{||\bar{w}_i||})] \pm \frac{5v_i}{d}\sqrt{\frac{\log(cd^2)}{c}} \qquad n > cd^2$$

$$= -\frac{v_i}{4}\Big[\phi\Big(\frac{w_{1i}+w_{2i}}{||\bar{w}_i||}\Big) - \phi\Big(\frac{w_{1i}-w_{2i}}{||\bar{w}_i||}\Big)\Big] \pm \frac{5v_i}{d}\sqrt{\frac{\log(cd^2)}{c}} \qquad \text{w.p. } 1-\frac{3}{n}$$

where (a) is due to $y_i x_{i2} = \mathbb{1}\{y_i=1\}\varepsilon_i$ & $\mathbb{1}\{y_j f(x_j) \leq 1\} = 1$ and (b) is due to $\mathbb{1}\{\bar{w}_i^T \bar{x}_j \geq k\} = \mathbb{1}\{||\bar{w}_i|| Z_j \geq k\}$. ■

Next, we provide a closed-form expression for the gradient along the noise coordinates:

Lemma D.7. *If $n > cd^2$ and $y_i f(x_i) < 1 \, \forall (x_i, y_i) \in \mathcal{S}_n$, then w.p. greater than $1 - \frac{1}{3n}$:*

$$\nabla_{\bar{w}_i} \mathcal{L}_f(\mathcal{S}_n) = \bar{\mathcal{G}} \bar{w}_i \pm \frac{3|v_i| \log(\sqrt{c}d)}{\sqrt{cd}} \frac{\bar{w}_i}{||\bar{w}_i||} \pm \frac{6|v_i|}{\sqrt{cd}} u_i^{\perp}$$
$$\bar{\mathcal{G}} = -\frac{v_i}{4||\bar{w}_i||} \Big[\varphi\Big(\frac{w_{1i} + w_{2i}}{||\bar{w}_i||} \Big) + \varphi\Big(\frac{w_{1i} - w_{2i}}{||\bar{w}_i||} \Big) - 2\varphi\Big(\frac{w_{1i}}{||\bar{w}_i||} \Big) \Big]$$

where $u_i^{\perp}$ is some unit vector orthogonal to $\bar{w}_i$.

Proof. Let $S \subset \mathbb{R}^{d-2}$ denote the subspace spanned by $\bar{w}_i$. Then, for any $x \in \mathbb{R}^d$, $x = x^S + x^{S^{\perp}}$ where x^S & $x^{S^{\perp}}$ are the orthogonal projections of x onto S and its orthogonal complement $S^{\perp}$. We show the ℓ_2 norm of the orthogonal projections of $\nabla_{\bar{w}_i} \mathcal{L}_f(\mathcal{S}_n)$ onto S and $S^{\perp}$ are $O(\frac{1}{\sqrt{d}})$:

$$\begin{aligned} \nabla_{\bar{w}_i} \mathcal{L}_f(\mathcal{S}_n) &= -\frac{v_i}{n} \sum_{j=1}^{n} \mathbb{1}\{y_j f(x_j) \le 1\} \mathbb{1}\{w_i^T x_j \ge 0\} y_j \bar{x}_j \\ &= \underbrace{-\frac{v_i}{n} \sum_{j=1}^{n} \mathbb{1}\{w_i^T x_j \ge 0\} y_j (\bar{x}_j^S)}_{\text{case 1}} \underbrace{-\frac{v_i}{n} \sum_{j=1}^{n} \mathbb{1}\{w_i^T x_j \ge 0\} y_j (\bar{x}_j^{S^{\perp}})}_{\text{case 2}} \end{aligned}$$

Next, we show that the projection of $\nabla_{\bar{w}_i} \mathcal{L}_f(\mathcal{S}_n)$ onto $S^{\perp}$ (i.e., case 2) has small norm w.p. greater than $1 - \frac{1}{d}$:

$$\begin{aligned} ||\nabla_{\bar{w}_i} \mathcal{L}_f(\mathcal{S}_n))^{S^{\perp}}|| &= ||\frac{v_i}{n} \sum_{j=1}^{n} \mathbb{1}\{w_i^T x_j \ge 0\} y_j \bar{x}_j^{S^{\perp}}|| = ||\frac{v_i}{n} \sum_{j=1}^{n} \mathbb{1}\{w_i^T x_j^S \ge 0\} y_j \bar{x}_j^{S^{\perp}}|| \\ &\overset{(a)}{\le} |v_i| \cdot ||\sum_{j=1}^{n} \mathcal{N}(0, \frac{1}{n^2} I_{d-2})|| = |v_i| \cdot ||\mathcal{N}(0, \frac{1}{n} I_{d-2})|| \\ &\overset{(b)}{\le} 4|v_i| \sqrt{\frac{d}{n}} \pm 2|v_i| \sqrt{\frac{\log n}{n}} \overset{(c)}{\le} \frac{6|v_i|}{\sqrt{cd}} \qquad \text{w.p. } 1 - \frac{1}{n} \end{aligned}$$

where (a) is because $x_j^S \perp \bar{x}^{S_j^{\perp}}$, (b) is via fact D.13 and (c) is due to $n \ge cd^2$. Next, we show that the norm of the gradient in the direction of $\bar{w}_i$ (i.e., case 1) is close to $\bar{\mathcal{G}}$ w.h.p.:

$$\begin{aligned} &\nabla_{\bar{w}_i} \mathcal{L}_f(\mathcal{S}_n))^S = -\frac{v_i}{n} \sum_{j=1}^{n} \mathbb{1}\{w_i^T x_j \ge 0\} y_j \bar{x}_j^S \\ &\overset{(a)}{=} -\Big(\frac{1}{n} \sum_{j=1}^{n} \mathbb{1}\{w_i^T x_j^S \ge 0\} y_j \bar{w}_i^T \bar{x}_j \Big) \frac{v_i \bar{w}_i}{||\bar{w}_i||^2} \\ &\overset{(b)}{=} -\Big(\frac{1}{n} \sum_{j=1}^{n} \mathbb{1}\Big\{ Z_j \ge \frac{-w_{1i} y - w_{2i} \mathbb{1}\{y_j = 1\} \varepsilon_j}{||\bar{w}_i||} \Big\} y_j Z_j \Big) \frac{v_i \bar{w}_i}{||\bar{w}_i||} \\ &= \Big(\sum_{l}^{\{0, \pm 1\}} \frac{(-1)^{\mathbb{1}\{l \ne 0\}}}{n} \sum_{i=1}^{n} \mathbb{1}\Big\{ x_{2j} = l \wedge Z_j \ge \frac{-w_{1i}(2l^2 - 1) - w_{2i} l}{||\bar{w}_i||} \Big\} Z_j \Big) \frac{v_i \bar{w}_i}{||\bar{w}_i||} \\ &= \Big[2\varphi(\frac{w_{i1}}{||\bar{w}_i||}) - \varphi(\frac{w_{1i} + w_{2i}}{||\bar{w}_i||}) - \varphi(\frac{w_{1i} - w_{2i}}{||\bar{w}_i||}) \pm \frac{5 \log n}{\sqrt{n}} \Big] \frac{v_i \bar{w}_i}{4||\bar{w}_i||} \qquad \text{via lemma D.11} \\ &= \bar{\mathcal{G}} \bar{w}_i \pm \frac{3|v_i| \log(\sqrt{c}d)}{\sqrt{cd}} \frac{\bar{w}_i}{||\bar{w}_i||} \qquad \text{w.p. } 1 - \frac{12}{n} \end{aligned}$$

where (a) is because $\bar{x}_j^S = \frac{\bar{w}_i^T x_j}{||\bar{w}_i||^2}\bar{w}_i$ and (b) is because (b) is due to $\mathbb{1}\{\bar{w}_i^T \bar{x}_j \geq k\} = \mathbb{1}\{||\bar{w}_i|| Z_j \geq k\}$. Therefore, by combining the results in case 1 and 2, the following holds w.p. greater than $1 - \frac{13}{n}$:

$$\nabla_{\bar{w}_i f}(\mathcal{S}_n) = \bar{\mathcal{G}}\bar{w}_i \pm \frac{3|v_i|\log(\sqrt{c}d)}{\sqrt{c}d}\frac{\bar{w}_i}{||\bar{w}_i||} \pm \frac{6|v_i|}{\sqrt{cd}}u_i^{\perp}$$

■

D.3 Miscellaneous Lemmas

Lemma D.8. *Let $X_i \sim \mathcal{N}(0, \sigma^2)$ and $\delta \in (0, 1)$. Then, $\max_{i\in[k]} |X_i| \leq \sigma\sqrt{2\log(\frac{2k}{\delta})}$ with probability greater than $1 - \delta$,*

Proof. Let φ denote the probability density function of the standard normal. Also let $Z \sim \mathcal{N}(0, 1)$. Then, for $t \geq 1$, we have:

$$\mathbb{P}(|X| \geq \sigma t) = \mathbb{P}(|Z| \geq t) = 2\int_t^\infty x\varphi(x)\,dx \leq \frac{2}{t}\int_t^\infty x\varphi(x)\,dx \overset{(a)}{\leq} \frac{2}{t}\int_\infty^t \varphi'(x)\,dx \leq \frac{2}{t}\varphi(t) \leq 2\varphi(t)$$

where (a) is because $\varphi'(x) = -x\varphi(x)$. Using union bound with $t = \sqrt{2\log(\frac{2k}{\delta})} \geq 1\,\forall\delta \in (0, 1)$ gives the desired result. ■

Lemma D.9. *Let ϕ and φ denote the cumulative distribution function and the probability density function of the standard gaussian. Then, for any $Z \sim \mathcal{N}(0, 1)$ and $k \in \mathbb{R}$:*

$$\mathbb{E}[\mathbb{1}\{Z \geq k\}Z] = \varphi(k) = \exp(-k^2/2)$$

Proof. The expectation $\mathbb{E}[\mathbb{1}\{Z \geq k\}Z]$ can be simplified as follows:

$$\mathbb{E}[\mathbb{1}\{Z \geq c\}Z] = \Pr[Z \geq c]\mathbb{E}\left[Z|Z \geq c\right] = \phi^c(k)\int_k^\infty x\frac{\varphi(x)}{\phi^c(k)}\,dx \overset{(a)}{=} -\int_k^\infty \varphi'(x)\,dx = \varphi(k)$$

where (a) is due to $\varphi'(x) = -x\varphi(x)$. ■

Lemma D.10. *Let $b_i \sim$ bernoulli(p) and $Z_i \sim \mathcal{N}(0, 1)$. Let $X_i = b_i\mathbb{1}\{Z_i \geq k\}$ and $\bar{X} = \frac{1}{n}\sum_{i=1}^n X_i$. Then:*

$$\Pr\left(|\bar{X} - p\phi(-k)| \geq \sqrt{\frac{\log n}{n}}\right) \leq \frac{1}{n}$$

Proof. Note that $\mathbb{E}\left[\bar{X}\right] = \mathbb{E}\left[X_i\right] = \mathbb{E}\left[b_i\right]\mathbb{E}\left[\mathbb{1}\{Z_i \geq k\}\right] = p\phi(-k)$ and $|X_i| \leq 1$. Therefore, using Hoeffding's inequality with $t = \sqrt{\frac{\log n}{n}}$ directly gives the result. ■

Lemma D.11. *Let $b_i \sim$ bern(p) and $Z_i \sim \mathcal{N}(0, 1)$. Let $X_i = b_i\mathbb{1}\{Z_i \geq k\}Z_i$ and $\bar{X} = \frac{1}{n}\sum_{i=1}^n X_i$. Then:*

$$\mathbb{P}\Big(|\bar{X} - p\varphi(k)| \leq \sqrt{\frac{2}{n}}\log n\Big) \geq 1 - \frac{4}{n}$$

Proof. Since $|X_i| = |b_i \mathbb{1}\{Z_i \geq k\} Z_i| \leq |Z_i|$, we have $\max_{i\in[n]} |X_i| \leq \sqrt{4\log(n)}$ w.p. at least $1 - \frac{2}{n}$ via lemma D.8. From lemma D.9, we get $\mathbb{E}[X_i] = \mathbb{E}[b_i]\,\mathbb{E}[\mathbb{1}\{Z_i \geq k\} Z_i] = p\varphi(k)$. Let $A = \mathbb{1}\left\{|X_i| \leq \sqrt{4\log(n)}\, \forall i \in [n]\right\}$. Given A, we can use Hoeffding's inequality with $t^* = \sqrt{\frac{2}{n}}\log n$ (and $\delta = 2/n$) to get the desired result, as follows:

$$\mathbb{P}(|\bar{X} - p\varphi(k)| \leq t^*) \geq \mathbb{P}(|\bar{X} - p\varphi(k)| \leq t^* \,|A)\mathbb{P}(A) \geq (1 - \frac{2}{n})^2 \geq 1 - \frac{4}{n}$$

Therefore, $\bar{X} = p\varphi(k) \pm \sqrt{\frac{2}{n}}\log n$ w.p. at least $1 - \frac{4}{n}$. ■

Lemma D.12. *Let $g : \mathbb{R}\backslash\{0\} \to \mathbb{R}$ be defined as $g_z(x) = \frac{1}{x}\exp(-\frac{z^2}{2x^2})$. Then, (1) $|z|$ and $-|z|$ are the global maximizer and minimizer respectively, and (2) g monotonically increases from $-|z|$ to $|z|$.*

Proof. Note that $g_z'(x) = \frac{1}{x^2}\exp(-\frac{z^2}{2x^2})(\frac{z^2}{x^2} - 1)$. Therefore, the critical points of g are $|z|$ and $-|z|$. Let $S = \{t : |t| \geq |z|, t \in \mathbb{R}/\{0\}\}$. Note that $g_z'(x) < 0$ for all $x \in S$ and $g_z'(x) > 0$ for all $x \in S^c$. Therefore, (1) and (2) hold. ■

Fact D.13. *Let $X \sim \mathcal{N}(0, \sigma^2 I_d)$ denote a d-dimensional gaussian vector. Then, from [39], w.p. greater than $1 - \delta$:*

$$||X||_2 \leq 4\sigma\sqrt{d} + 2\sigma$$

References

[1] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*, 2018.

[2] Peter L Bartlett. The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. *IEEE transactions on Information Theory*, 44(2):525–536, 1998.

[3] Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*, pages 6240–6249, 2017.

[4] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248*, 2017.

[5] Alon Brutzkus, Amir Globerson, Eran Malach, and Shai Shalev-Shwartz. Sgd learns over-parameterized networks that provably generalize on linearly separable data. *arXiv preprint arXiv:1710.10174*, 2017.

[6] Jacob Buckman, Aurko Roy, Colin Raffel, and Ian Goodfellow. Thermometer encoding: One hot way to resist adversarial examples. 2018.

[7] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. arxiv e-prints, page. *arXiv preprint arXiv:1608.04644*, 2, 2016.

[8] Lenaic Chizat and Francis Bach. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. *arXiv preprint arXiv:2002.04486*, 2020.

[9] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[10] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Xiaolin Hu, J Li, and J Zhu. Boosting adversarial attacks with momentum. arxiv preprint. *arXiv preprint arXiv: 1710.06081*, 2017.

[11] Gintare Karolina Dziugaite and Daniel M Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv preprint arXiv:1703.11008*, 2017.

[12] Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. Exploring the landscape of spatial robustness. *arXiv preprint arXiv:1712.02779*, 2017.

[13] Noah Golowich, Alexander Rakhlin, and Ohad Shamir. Size-independent sample complexity of neural networks. *arXiv preprint arXiv:1712.06541*, 2017.

[14] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[15] Suriya Gunasekar, Jason D Lee, Daniel Soudry, and Nati Srebro. Implicit bias of gradient descent on linear convolutional networks. In *Advances in Neural Information Processing Systems*, pages 9461–9471, 2018.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. corr abs/1512.03385 (2015), 2015.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[18] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

[19] Gao Huang, Zhuang Liu, and Kilian Q Weinberger. Densely connected convolutional networks. corr abs/1608.06993 (2016). *arXiv preprint arXiv:1608.06993*, 2016.

[20] Ziwei Ji and Matus Telgarsky. The implicit bias of gradient descent on nonseparable data. In *Conference on Learning Theory*, pages 1772–1798, 2019.

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[22] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. arxiv e-prints, page. *arXiv preprint arXiv:1612.01474*, 5, 2016.

[23] Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*, 2017.

[24] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3, 2013.

[25] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[26] Karttikeya Mangalam and Vinay Uday Prabhu. Do deep neural networks learn shallow learnable examples first? 2019.

[27] Vaishnavh Nagarajan and J Zico Kolter. Uniform convergence may be unable to explain generalization in deep learning. In *Advances in Neural Information Processing Systems*, pages 11611–11622, 2019.

[28] Preetum Nakkiran, Gal Kaplun, Dimitris Kalimeris, Tristan Yang, Benjamin L Edelman, Fred Zhang, and Boaz Barak. Sgd on neural networks learns functions of increasing complexity. *arXiv preprint arXiv:1905.11604*, 2019.

[29] Behnam Neyshabur, Srinadh Bhojanapalli, and Nathan Srebro. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. *arXiv preprint arXiv:1707.09564*, 2017.

[30] Nicolas Papernot, Patrick D McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. corr abs/1511.04508 (2015). In *37th IEEE Symposium on Security and Privacy*, 2015.

[31] Jie Ren, Peter J Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark Depristo, Joshua Dillon, and Balaji Lakshminarayanan. Likelihood ratios for out-of-distribution detection. In *Advances in Neural Information Processing Systems*, pages 14680–14691, 2019.

[32] Kevin Roth, Yannic Kilcher, and Thomas Hofmann. The odds are odd: A statistical test for detecting adversarial examples. *arXiv preprint arXiv:1902.04818*, 2019.

[33] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[34] Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! In *Advances in Neural Information Processing Systems*, pages 3353–3364, 2019.

[35] Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878, 2018.

[36] Nitish Srivastava. Improving neural networks with dropout. *University of Toronto*, 182(566):7, 2013.

[37] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[38] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.

[39] Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

[40] Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. *arXiv preprint arXiv:2001.03994*, 2020.

[41] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 501–509, 2019.

[42] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*, 2017.

[43] Valentina Zantedeschi, Maria-Irina Nicolae, and Ambrish Rawat. Efficient defenses against adversarial attacks. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 39–49, 2017.

[44] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P Xing, Laurent El Ghaoui, and Michael I Jordan. Theoretically principled trade-off between robustness and accuracy. *arXiv preprint arXiv:1901.08573*, 2019.