Gradient Starvation: A Learning Proclivity in Neural Networks

Anonymous Author(s)
Affiliation
Address
email

Abstract

We identify and formalize a fundamental gradient descent phenomenon leading to a learning proclivity in over-parameterized neural networks. *Gradient Starvation* arises when cross-entropy loss is minimized by capturing only a subset of features relevant for the task, despite the presence of other predictive features that fail to be discovered. This work provides a theoretical explanation for the emergence of such feature imbalances in neural networks. Using tools from Dynamical Systems theory, we identify simple properties of learning dynamics during gradient descent that lead to this imbalance, and prove that such a situation can be expected given certain statistical structure in training data. Based on our proposed formalism, we develop guarantees for a novel but simple regularization method aimed at decoupling feature learning dynamics, improving accuracy and robustness in cases hindered by gradient starvation. We illustrate our findings with simple and real-world out-of-distribution (OOD) generalization experiments.

1 Introduction

In 1904, a horse named *Hans* attracted worldwide attention due to the belief that it was capable of doing arithmetic calculations [79]. Its trainer would ask Hans a question, and Hans would reply by tapping on the ground with its hoof. However, it was later revealed that the horse was only noticing subtle but distinctive signals in its trainer's unconscious behavior, unbeknown to him, and not actually performing arithmetic. An analogous phenomenon has been noticed when training neural networks [e.g. 83, 104, 50, 36, 17, 13, 34, 47, 102, 74, 44, 19, 58, 75]. In many cases, state-of-the-art neural networks appear to focus on low-level **superficial correlations**, rather than more abstract and robustly informative features of interest [16, 85, 37, 66, 27]. Neural networks learning to classify images only based on watermarks or latching on data compression artifacts are among the authors' personal experiences.

The rationale behind this phenomenon is well known by practitioners: given strongly-correlated and fast-to-learn features in training data, gradient descent is biased towards learning them first. However, the precise conditions leading to such learning dynamics, and how one might intervene to control this *feature imbalance* are not entirely understood. Recent work aims at identifying the reasons behind this phenomenon [94, 68, 21, 71, 47, 74, 97, 89, 81, 100, 39, 77, 4], while complementary work quantifies resulting shortcomings, including poor generalization to out-of-distribution (OOD) test data, reliance upon spurious correlations, and lack of robustness [27, 66, 75, 38, 61, 62, 8]. However most established work focuses on squared-error loss and its particularities, where results do not readily generalize to other objective forms. This is especially problematic since for several classification applications, cross-entropy is the loss function of choice, yielding very distinct learning dynamics. In this paper, we argue that *Gradient Starvation*, first coined in [24], is a leading cause for this *feature imbalance* in neural networks trained with cross-entropy, and propose a simple approach

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

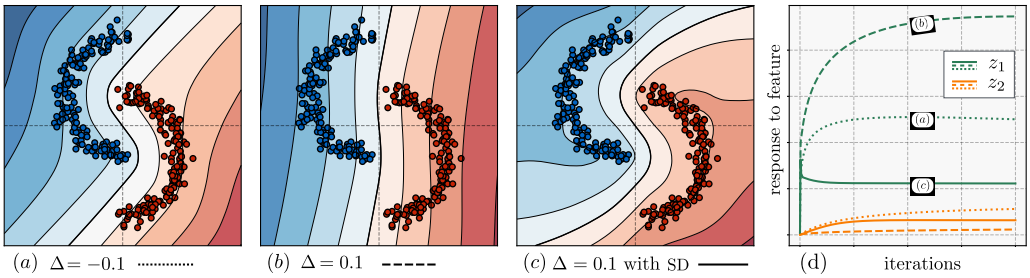

Figure 1: Diagram illustrating the effect of gradient starvation in a simple 2-D classification task. **(a)** Data is not linearly separable and the learned decision boundary is curved. **(b)** Data is linearly separable by a small margin ($\Delta = 0.1$). This small margin allows the network to discriminate confidently only along the horizontal axis and ignore the vertical axis. **(c)** Data is linearly separable as in (b). However, with the proposed Spectral decoupling (SD), a curved decision boundary with a large margin is learned. **(d)** Diagram shows the evolution of two of the features (Eq. 4) of the dynamics in three cases shown as dotted, dashed and solid lines. **Analysis:** (dotted) vs (dashed): Linear separability of the data results in an increase in z_1 and a decrease (starvation) of z_2. (dashed) vs (solid): SD suppresses z_1 and hence allows z_2 to grow. Decision boundaries are averaged over ten runs. More experiments with common regularization methods are provided in SM. B.

to mitigate it.

Here we summarize our contributions:

- We provide a theoretical framework to study the learning dynamics of wide, over-parameterized neural networks trained with cross-entropy loss.
- Using perturbation analysis, we formalize Gradient Starvation (GS) in view of the coupling between the dynamics of orthogonal directions in the feature space (Thm. 2).
- We leverage our theory to introduce Spectral Decoupling (SD) (Eq. 17) and prove this simple regularizer helps to decouple learning dynamics, mitigating GS.
- We support our findings with extensive empirical results on a variety of classification and adversarial attack tasks. All code and experiment details available at `GitHub repository`.

In the rest of the paper, we first present a simple example to outline the consequences of GS. We then present our theoretical results before outlining a number of numerical experiments. We close with a review of related work followed by a discussion.

2 Gradient Starvation: A simple example

Consider a 2-D classification task with a training set consisting of two classes, as shown in Figure 1. A two-layer ReLU network with 500 hidden units is trained with cross-entropy loss for two different arrangements of the training points. The difference between the two arrangements is that, in one setting, the data is not linearly separable, but a slight shift makes it linearly separable in the other setting. This small shift allows the network to achieve a negligible loss by only learning to discriminate along the horizontal axis, ignoring the other. This contrasts with the other case, where both features contribute to the learned classification boundary, which arguably matches the data structure better. We observe that training longer or using different regularizers, including weight decay [55], dropout [92], batch normalization [45], as well as changing the optimization algorithm to Adam [52] or changing the network architecture or the coordinate system, do not encourage the network to learn a curved decision boundary. (See SM. B for more details.)

We argue that this occurs because cross-entropy loss leads to gradients "starved" of information from vertical features. Simply put, when one feature is learned faster than the others, the gradient contribution of examples containing that feature is diminished (i.e., they are correctly processed based on that feature alone). This results in a lack of sufficient gradient signal, and hence prevents any remaining features from being learned. This simple mechanism has potential consequences, which we outline below.

Lack of robustness. In the example above, even in the right plot, the training loss is nearly zero, and the network is very confident in its predictions. However, the decision boundary is located very

close to the data points. This could lead to adversarial vulnerability as well as lack of robustness when generalizing to out-of-distribution data.

Excessive invariance. GS could also result in neural networks that are invariant to task-relevant changes in the input. In the example above, it is possible to obtain a data point with low probability under the data distribution, but that would still be classified with high confidence.

Implicit regularization. One might argue that according to Occam's razor, a simpler decision boundary should generalize better. In fact, if both training and test sets share the same dominant feature (in this example, the feature along the horizontal axis), GS naturally prevents the learning of less dominant features that could otherwise result in overfitting. Therefore, depending on our assumptions on the training and test distributions, GS could also act as an implicit regularizer. We provide further discussion on the *implicit regularization* aspect of GS in Section 5.

3 Theoretical Results

In this section, we study the learning dynamics of neural networks trained with cross-entropy loss. Particularly, we seek to decompose the learning dynamics along orthogonal directions in the feature space of neural networks, to provide a formal definition of GS, and to derive a simple regularization method to mitigate it. For analytical tractability, we make three key assumptions: (1) we study deep networks in the Neural Tangent Kernel (NTK) regime, (2) we treat a binary classification task, (3) we decompose the interaction between two features. In Section 4, we demonstrate our results hold beyond these simplifying assumptions, for a wide range of practical settings. All derivation details can be found in SM C.

3.1 Problem Setup and Gradient Starvation Definition

Let $\mathcal{D} = \{\mathbf{X}, \mathbf{y}\}$ denote a training set containing n datapoints with d dimensions, where, $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_n] \in \mathbb{R}^{n \times d}$ and their corresponding class label $\mathbf{y} \in \{-1, +1\}^n$. Also let $\hat{\mathbf{y}}(\mathbf{X}) := f^{(L)}(\mathbf{X}) : \mathbb{R}^{n \times d} \to \mathbb{R}^n$ represent the logits of an L-layer fully-connected neural network where each hidden layer $h^{(l)}(x) \in \mathbb{R}^{d_l}$ is defined as follows,

$$\begin{cases} f^{(l)}(\mathbf{x}_i) = \mathbf{W}^{(l)} h^{(l-1)}(\mathbf{x}_i) \\ h^{(l)}(\mathbf{x}_i) = \sqrt{\frac{\gamma}{d_l}} \xi(f^{(l)}(\mathbf{x}_i)) \end{cases}, \; l \in \{0, 1, ..., L\}, \tag{1}$$

in which $\mathbf{W}^{(l)} \in \mathbb{R}^{d_l \times d_{l-1}}$ is a weight matrix drawn from $\mathcal{N}(0, \mathbf{I})$ and γ is a scaling factor to ensure that norm of each $h^{(l-1)}$ is preserved at initialization (See [25] for a formal treatment). The function $\xi(.)$ is also an element-wise non-linear activation function.

Let $\boldsymbol{\theta} = \text{concat}\big(\cup_{l=1}^{L} \text{vec}(\mathbf{W}^{(l)})\big) \in \mathbb{R}^m$ be the concatenation of all vectorized weight matrices with m as the total number of parameters. In the NTK regime [48], in the limit of infinite width, the output of the neural network can be approximated as a linear function of its parameters governed by the neural tangent random feature (NTRF) matrix [22],

$$\mathbf{\Phi}(\mathbf{X}, \boldsymbol{\theta}) = \frac{\partial \hat{\mathbf{y}}(\mathbf{X}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \in \mathbb{R}^{n \times m}. \tag{2}$$

In the wide-width regime, the NTRF changes very little during training [60], and the output of the neural network can be approximated by a first order Taylor expansion around the initialization parameters. Setting $\mathbf{\Phi}_0 \equiv \mathbf{\Phi}(\mathbf{X}, \boldsymbol{\theta}_0)$ and then, without loss of generality, centering parameters and the output coordinates to their value at the initialization ($\boldsymbol{\theta}_\mathbf{0}$ and $\hat{\mathbf{y}}_0$), we get

$$\hat{\mathbf{y}}(\mathbf{X}, \boldsymbol{\theta}) = \mathbf{\Phi}_0 \boldsymbol{\theta}. \tag{3}$$

Dominant directions in the feature space as well as the parameter space are given by principal components of the NTRF matrix $\mathbf{\Phi}_0$, which are the same as those of the NTK Gram matrix [101]. We therefore introduce the following definition.

Definition 1 (Features and Responses)**.** *Consider the singular value decomposition (SVD) of the matrix* $\mathbf{Y}\mathbf{\Phi}_0 = \mathbf{U}\mathbf{S}\mathbf{V}^T$*, where* $\mathbf{Y} = diag(\mathbf{y})$*. The* j*th* **feature** *is given by* $(\mathbf{V}^T)_{j.}$*. The* **strength** *of* j*th feature is represented by* $s_j = (\mathbf{S})_{jj}$*. Also,* $(\mathbf{U})_{.j}$ *contains the weights of this feature in all examples. A neural network's* **response to a feature** j *is given by*

$$\mathbf{z} := \mathbf{U}^T \mathbf{Y} \hat{\mathbf{y}} = \mathbf{S}\mathbf{V}^T \boldsymbol{\theta}. \tag{4}$$

In Eq. 4, the response to feature j is the sum of the responses to every example in $(\mathbf{Y}\hat{\mathbf{y}})$ multiplied by the weight of the feature in that example $(\mathbf{U}^T)$. For example, if all elements of $(\mathbf{U})_{.j}$ are positive, it indicates a perfect correlation between this feature and class labels. We are now equipped to formally define GS.

Definition 2 (Gradient Starvation)**.** *Let z_j^* denote system 3's response to feature j at training optimum $\boldsymbol{\theta}^*$. Feature i* **starves the gradient** *for feature j if $dz_j^*/d(s_i^2) < 0$.*

This definition of GS implies that an increase in the strength of feature i has a detrimental effect on the learning of feature j. We now derive conditions for which learning dynamics of system 3 suffer from GS.

3.2 Training Dynamics

We consider the widely used ridge-regularized cross-entropy loss function,

$$\mathcal{L}(\boldsymbol{\theta}) = \mathbf{1} \cdot \log\left[1 + \exp\left(-\mathbf{Y}\hat{\mathbf{y}}\right)\right] + \frac{\lambda}{2}\|\boldsymbol{\theta}\|^2, \tag{5}$$

where $\mathbf{1}$ is a vector of size n with all its elements equal to 1. This vector form simply represents a summation over all the elements of the vector it is multiplied to. $\lambda \in [0, \infty)$ denotes the weight decay coefficient.

Direct minimization of this loss function using the gradient descent obeys coupled dynamics and is difficult to treat directly [24]. To overcome this problem, we call on a variational approach that leverages the Legendre transformation of the loss function. This allows tractable dynamics that can directly incorporate rates of learning in different feature directions. Following [46], we note the following inequality,

$$\log\left[1 + \exp\left(-\mathbf{Y}\hat{\mathbf{y}}\right)\right] \geq H(\boldsymbol{\alpha}) - \boldsymbol{\alpha} \odot \mathbf{Y}\hat{\mathbf{y}}, \tag{6}$$

where $H(\boldsymbol{\alpha}) = -\left[\boldsymbol{\alpha}\log\boldsymbol{\alpha} + (1-\boldsymbol{\alpha})\log(1-\boldsymbol{\alpha})\right]$ is Shannon's binary entropy function and $\boldsymbol{\alpha} \in (0,1)^n$ is a variational parameter defined for each training example. Crucially, the equality holds when the maximum of r.h.s. w.r.t $\boldsymbol{\alpha}$ is achieved at $\boldsymbol{\alpha}^* = \frac{\partial \mathcal{L}}{\partial (\mathbf{Y}\hat{\mathbf{y}})^T}$, which leads to the following optimization problem,

$$\min_{\boldsymbol{\theta}} \mathcal{L}(\theta) = \min_{\boldsymbol{\theta}} \max_{\boldsymbol{\alpha}} \left(\mathbf{1} \cdot H(\boldsymbol{\alpha}) - \boldsymbol{\alpha}\mathbf{Y}\hat{\mathbf{y}} + \frac{\lambda}{2}\|\boldsymbol{\theta}\|^2\right), \tag{7}$$

where the order of min and max can be swapped (see Lemma 3 of [46]). Since the neural network's output is approximated by a linear function of $\boldsymbol{\theta}$, the minimization can be performed analytically with an optimal value $\boldsymbol{\theta}^{*T} = \frac{1}{\lambda}\boldsymbol{\alpha}\mathbf{Y}\boldsymbol{\Phi}_0$, given by a weighted sum of the training examples. This results in the following maximization problem on the dual variable, i.e., $\min_\theta \mathcal{L}(\theta)$ is equivalent to,

$$\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) = \max_{\boldsymbol{\alpha}} \left(\mathbf{1} \cdot H(\boldsymbol{\alpha}) - \frac{1}{2\lambda}\boldsymbol{\alpha}\mathbf{Y}\boldsymbol{\Phi}_0\boldsymbol{\Phi}_0^T\mathbf{Y}^T c^T\right). \tag{8}$$

By applying continuous-time gradient ascent on this optimization problem, we derive an autonomous differential equation for the evolution of $\boldsymbol{\alpha}$, which can be written in terms of features (see Definition 1),

$$\dot{\boldsymbol{\alpha}} = \eta\left(\log\frac{\mathbf{1}-\boldsymbol{\alpha}}{\boldsymbol{\alpha}} - \frac{1}{\lambda}\boldsymbol{\alpha}\mathbf{U}\mathbf{S}^2\mathbf{U}^T\right), \tag{9}$$

where η is the learning rate (see SM C.1 for more details). For this dynamical system, we see that the logarithm term acts as barriers that keep $\alpha_i \in (0,1)$. The other term depends on the matrix $\mathbf{U}\mathbf{S}^2\mathbf{U}^T$, which is positive definite, and thus pushes the system towards the origin and therefore drives learning.

When $\lambda \ll s_k^2$, where k is an index over the singular values, the linear term dominates Eq. 9, and the fixed point is drawn closer towards the origin. Approximating dynamics with a first order Taylor expansion around the origin of the second term in Eq. 9, we get

$$\dot{\boldsymbol{\alpha}} \approx \eta\left(-\log\boldsymbol{\alpha} - \frac{1}{\lambda}\boldsymbol{\alpha}\mathbf{U}\left(\mathbf{S}^2 + \lambda\mathbf{I}\right)\mathbf{U}^T\right), \tag{10}$$

with stability given by the following theorem with proof in SM C.

Theorem 1. *Any fixed points of the system in Eq. 10 is attractive in the domain* $\alpha_i \in (0, 1)$.

At the fixed point $\boldsymbol{\alpha}^*$, corresponding to the optimum of Eq. 8, the feature response of the neural network is given by,

$$\mathbf{z}^* = \frac{1}{\lambda}\mathbf{S}^2\mathbf{U}^T\boldsymbol{\alpha}^{*T}. \tag{11}$$

Below, we study how the strength of one feature could impact the response of the network to another feature which leads to GS.

3.3 Gradient Starvation Regime

In general, we do not expect to find an analytical solution for the dynamics of the coupled non-linear dynamical system of Eq. 10. However, there are at least two cases where a decoupled form for the dynamics allows to find an exact solution. We first introduce these cases and then study their perturbation to outline general lessons.

1. If the matrix of singular values $\mathbf{S}^2$ is proportional to the identity: This is the case where all the features have the same strength s^2. The fixed points are then given by,

$$\alpha_i^* = \frac{\lambda\mathcal{W}(\lambda^{-1}s^2+1)}{s^2+\lambda}, \qquad z_j^* = \frac{s^2\mathcal{W}(\lambda^{-1}s^2+1)}{s^2+\lambda}\sum_i u_{ij}, \tag{12}$$

where $\mathcal{W}$ is the Lambert W function.

2. If the matrix $\mathbf{U}$ is a permutation matrix: This is the case in which each feature is associated with a single example only. The fixed points are then given by,

$$\alpha_i^* = \frac{\lambda\mathcal{W}(\lambda^{-1}s_i^2+1)}{s_i^2+\lambda}, \qquad z_j^* = \frac{s_i^2\mathcal{W}(\lambda^{-1}s_i^2+1)}{s_i^2+\lambda}. \tag{13}$$

To study a minimal case of starvation, we consider a variation of case 2 with the following assumption which implies that each feature is not associated with a single example anymore.

Assumption 1. *Assume* $\mathbf{U}$ *is a perturbed identity matrix in which the off-diagonal elements are proportional to a small parameter* $\delta > 0$.

Lemma 1. *Under assumption 1, the fixed point of the dynamical system in Eq. 10 can be approximated by,*

$$\boldsymbol{\alpha}^* = (1 - \log(\boldsymbol{\alpha}_0^*))\left[\mathbf{A} + \mathit{diag}\left(\boldsymbol{\alpha}_0^{*-1}\right)\right]^{-1}, \tag{14}$$

where $\mathbf{A} = \lambda^{-1}\mathbf{U}(\mathbf{S^2} + \lambda\mathbf{I})\mathbf{U}^T$ *and* $\boldsymbol{\alpha}_0^*$ *is the fixed point of the uncoupled system with* $\delta = 0$.

For sake of ease of derivations, we consider the two dimensional case where,

$$\mathbf{U} = \begin{pmatrix} \sqrt{1-\delta^2} & -\delta \\ \delta & \sqrt{1-\delta^2} \end{pmatrix}, \tag{15}$$

which is equivalent to a U matrix with two blocks of features with no intra-block coupling and δ amount of inter-block coupling.

Theorem 2 (Gradient Starvation Regime)**.** *Consider a neural network in the linear regime, trained under cross-entropy loss for a binary classification task. With definition 1, assuming coupling between features 1 and 2 as in Eq. 15 and* $s_1^2 > s_2^2$*, we have,*

$$\frac{\mathrm{d}z_2^*}{\mathrm{d}s_1^2} < 0, \tag{16}$$

which implies GS.

While Thm. 2 outlines conditions for GS in two dimensional feature space, we note that the same rationale naturally extends to higher dimensions, where GS is defined pairwise over feature directions. For a classification task, Thm. 2 indicates that gradient starvation occurs when the data admits different feature strengths, and coupled learning dynamics. GS is thus naturally expected with cross-entropy loss. Its detrimental effects however (as outlined in Sect. 2) arise in settings with large discrepancies between feature strengths, along with network connectivity that couples these features' directions. This phenomenon readily extends to multi-class settings, and we validate this case with experiments in Sect. 4. Next, we introduce a simple regularizer that encourages feature decoupling, thus mitigating GS by insulating strong features from weaker ones.

3.4 Spectral Decoupling

By tracing back the equations of the previous section, one may realize that the term $U^T S^2 U$ in Eq. 9 is not diagonal in the general case, and consequently introduces coupling between α_i's and hence, between the features z_i's. We would like to discourage solutions that couple features in this way. To that end, we introduce a simple regularizer: Spectral Decoupling (SD). SD replaces the general L2 weight decay term in Eq. 5 with an L2 penalty exclusively on the network's logits, yielding

$$\mathcal{L}(\boldsymbol{\theta}) = \mathbf{1} \cdot \log\left[1 + \exp\left(-\mathbf{Y}\hat{\mathbf{y}}\right)\right] + \frac{\lambda}{2}\|\hat{\mathbf{y}}\|^2. \tag{17}$$

Repeating the same analysis steps taken above, but with SD instead of general L2 penalty, the optimal value for $\boldsymbol{\theta}^*$ becomes $\boldsymbol{\theta}^* = \frac{1}{\lambda}\boldsymbol{\alpha}\boldsymbol{Y}\boldsymbol{\Phi}_0 V \mathbf{S}^{-2} V^T$. This new expression for $\boldsymbol{\theta}^*$ results in the following modification of Eq. 9,

$$\dot{\boldsymbol{\alpha}} = \eta\left(\log\frac{\mathbf{1}-\boldsymbol{\alpha}}{\boldsymbol{\alpha}} - \frac{1}{\lambda}\boldsymbol{\alpha}\mathbf{U}\mathbf{S}^2\mathbf{S}^{-2}\mathbf{U}^T\right) = \eta\left(\log\frac{\mathbf{1}-\boldsymbol{\alpha}}{\boldsymbol{\alpha}} - \frac{1}{\lambda}\boldsymbol{\alpha}\right), \tag{18}$$

where as earlier, $\log$ and division are taken element-wise on the coordinates of $\boldsymbol{\alpha}$.

Note that in contrast to Eq. 9 the matrix multiplication involving U and S in Eq. 18 cancels out, leaving α_i independent of other $\alpha_{j\neq i}$'s. We point out this is true for any initial coupling, without simplifying assumptions. Thus, a simple penalty on output weights promotes decoupled dynamics across the dual parameter α_i's, which track learning dynamics of feature responses (see Eq. 7). Together with Thm. 2, Eq. 18 suggests SD should mitigate GS and promote balanced learning dynamics across features. We now verify this in numerical experiments.

4 Experiments

The experiments presented here are designed to outline the presence of GS and its consequences, as well as the efficacy of our proposed regularization method to alleviate them. Consequently, we highlight that achieving state-of-the-art results is not the objective. For more details, see SM. B.

4.1 Two-Moon classification and the margin

Recall the simple 2-D classification task between red and blue data points in Fig. 1. Fig. 1 **(c)** demonstrates the learned decision boundary when SD is used. SD leads to learning a curved decision boundary with a larger margin in the input space. See SM. B for additional details and experiments.

4.2 CIFAR classification and adversarial robustness

To study the classification margin in deeper networks, we conduct a classification experiment on CIFAR-10, CIFAR-100, and CIFAR-2 (cats vs dogs of CIFAR-10) [54] using a convolutional network with ReLU non-linearity. Unlike linear models, the margin to a non-linear decision boundary cannot be computed analytically. Therefore, following the approach in [70], we use "the norm of input-disturbance required to cross the decision boundary" as a proxy for the margin.

Dataset	Method	Train*	Test IID	Test OOD†
Cifar-2	w/o SD	100.0% ± 0.0	95.2% ± 0.12	42.3% ± 3.0
	w/ SD (λ=0.01)	100.0% ± 0.0	95.3% ± 0.17	69.7% ± 2.9
Cifar-10	w/o SD	99.9% ± 0.01	92.8% ± 0.15	30.1% ± 2.1
	w/ SD (λ=0.01)	99.9% ± 0.01	92.9% ± 0.16	67.7% ± 1.5
Cifar-100	w/o SD	99.7% ± 0.01	69.2% ± 0.29	14.3% ± 2.0
	w/ SD (λ=0.05)	99.7% ± 0.02	70.5% ± 0.26	24.9% ± 1.9

† Accuracy (± std) for 10 runs.

Table 1: Table compares adversarial robustness of ERM (vanilla cross-entropy) vs SD with a CNN trained on CIFAR-2, 10, and 100 (setup of [70]). SD consistently achieves a better OOD performance.

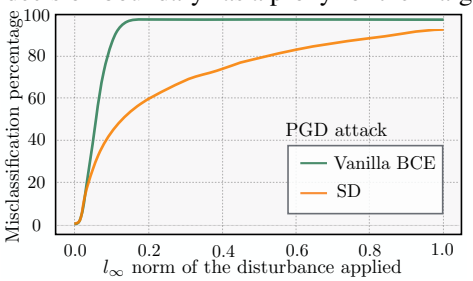

Figure 2: The plot shows the cumulative distribution function (CDF) of the margin for the CIFAR-2 binary classification. SD appears to improve the margin considerably.

The disturbance on the input is computed by projected gradient descent (PGD) [82], a well-known adversarial attack.

Table 1 includes the results for IID (original test set) and OOD (perturbed test set by $\epsilon_{\text{PGD}} = 0.05$). Fig. 2 shows the percentage of mis-classifications as the norm of disturbance is increased for

the Cifar-2 dataset. This plot can be interpreted as the cumulative distribution function (CDF) of the margin and hence a lower curve reads as a more robust network with a larger margin. This experiment suggests that when trained with vanilla cross-entropy, even slight disturbances in the input deteriorates the network's classification accuracy. That is while spectral decoupling (SD) improves the margin considerably. Importantly, this improvement in robustness does not seem to compromise the noise-free test performance. It should also be highlighted that SD does not explicitly aim at maximizing the margin and the observed improvement is in fact a by-product of decoupled learning of latent features.

4.3 Colored MNIST with color bias

We conduct experiments on the Colored MNIST Dataset, proposed in [8]. The task is to predict binary labels $y = -1$ for digits 0 to 4 and $y = +1$ for digits 5 to 9. A color channel (red, green) is artificially added to each example to deliberately impose a spurious correlation between the color and the label. The task has three environments:

- Training env. 1: Color is correlated with the labels with 0.9 probability.
- Training env. 2: Color is correlated with the labels with 0.8 probability.
- Testing env.: Color is correlated with the labels with 0.1 probability (0.9 reversely correlated).

Because of the opposite correlation between the color and the label in the test set, only learning to classify based on color would be disastrous at testing. For this reason, Empirical Risk Minimization (ERM - vanilla cross-entropy) performs very poorly on the test set (23.7 % accuracy) as shown in Tab. 2.

Method	**Train Accuracy**	**Test Accuracy**
ERM (Vanilla Cross-Entropy)	91.1 % (±0.4)	23.7 % (±0.8)
REx [56]	71.5 % (±1.0)	68.7 % (±0.9)
IRM [8]	70.5 % (±0.6)	67.1 % (±1.4)
SD (this work)	70.0 % (±0.9)	68.4 % (±1.2)
Oracle - (grayscale images)	73.5 % (±0.2)	73.0 % (±0.4)
Random Guess	50 %	50 %

Table 2: Test accuracy on test examples of the Colored MNIST after training for 1k epochs. The standard deviation over 10 runs is reported in parenthesis. ERM stands for the empirical risk minimization. Oracle is an ERM trained on grayscale images. Note that due to 25 % label noise, a hypothetical optimum achieves 75 % accuracy (the upper bound).

Invariant Risk Minimization (IRM) [8] on the other hand, performs well on the test set with (67.1 % accuracy). However, IRM requires access to multiple (two in this case) separate training environments with varying amount of spurious correlations. IRM uses the variance between environments as a signal for learning to be "invariant" to spurious correlations.

Despite simplicity, our spectral penalty (SD) also achieves an accuracy of 68.4 %. Its performance is remarkable as SD does not require access to multiple environments and yet performs well when trained on a single environment (in this case the aggregation of both of the training environments).

A natural question that arises is **"How does SD learn to ignore the color feature without having access to multiple environments?"** The short answer is that **it does not**! In fact, we argue that SD learns the color feature but it **also** learns other predictive features, i.e., the digit shape features. At test time, the predictions resulting from the shape features prevail over the color feature. To validate this hypothesis, we study a trained model with each of these methods (ERM, IRM, SD) on <u>four variants of the test environment</u>: 1) grayscale-digits: No color channel is provided and the network should rely on shape features only. 2) colored-digits: Both color and digit are provided however the color is negatively correlated (opposite of the training set) with the label. 3) grayscale-blank: All images are grayscale and blank and hence do not provide any information. 4) colored-blank: Digit features are removed and only the color feature is kept, also with reverse label compared to training. Fig. 3 summarizes the results. For more discussions see SM B

As a final remark, we should highlight that, by design, this task assumes access to the test environment for hyperparameter tuning for all the reported methods. This is not a valid assumption in general, and hence the results should be only interpreted as a probe that shows that SD could provide an important level of control over what features are learned.

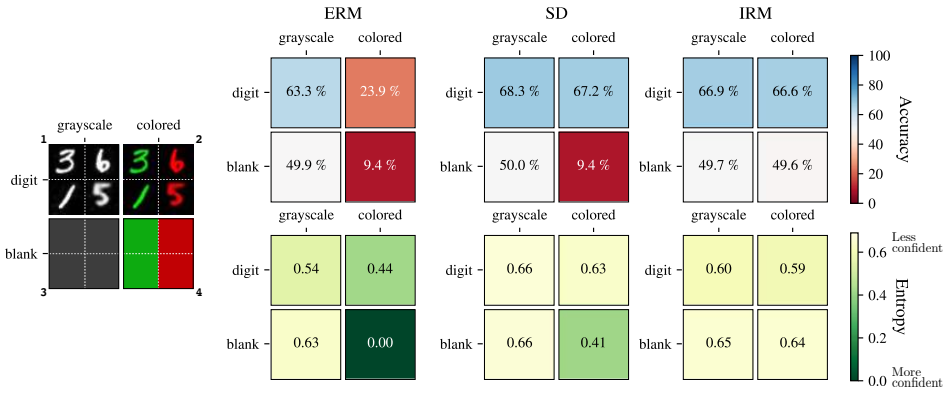

Figure 3: Diagram comparing ERM, SD, and IRM on four different test environments on which we evaluate a pre-trained model. Top and bottom rows show the accuracy and the entropy (inverse of confidence), respectively. **Analysis:** Compare three values of 9.4 %, 9.4 %, and 49.6 % : Both ERM and SD have learned the color feature but since it is inversely correlated with the label, when only the color feature is provided, as expected both ERM and SD performs poorly. Now compare 0.00 and 0.41 : Although both ERM and SD have learned the color feature, ERM is much more confident on its predictions (zero entropy). As a consequence, when digit features are provided along with the color feature (colored-digit environment), ERM still performs poorly (23.9 %) but SD achieves significantly better results (67.2 %). IRM ignores the color feature altogether but it requires access to multiple training environments.

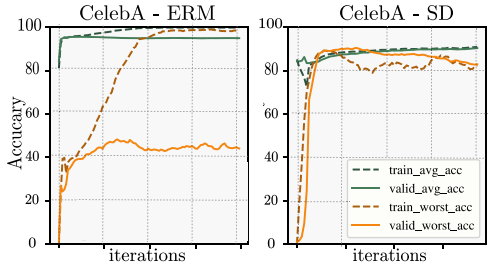

Figure 4: CelebA: `blond` vs `dark` hair classification. The `HairColor` and the `Gender` are spuriously correlated which leads to poor OOD performance with ERM, but SD significantly improves performance. ERM's worst group accuracy is significantly lower than SD.

Method	Average Acc.	Worst Group Acc.
ERM	94.61 % (±0.67)	40.35 % (±1.68)
SD (this work)	91.64 % (±0.61)	**83.24** % (±2.01)
LfF	N/A	81.24 % (±1.38)
Group DRO*	91.76 % (±0.28)	87.78 % (±0.96)

Table 3: CelebA: `blond` vs `dark` hair classification with spurious correlation. We report test performance over ten runs. SD significantly improves upon ERM. *Group DRO [86] requires explicit information about the spurious correlation. LfF [69] requires simultaneous training of two networks.

4.4 CelebA with gender bias

The CelebA dataset [63] contains 162k celebrity faces with binary attributes associated with each image. Following the setup of [86], the task is to classify images with respect to their hair color into two classes of blond or dark hair. However, the `Gender` $\in$ {`Male`, `Female`} is spuriously correlated with the `HairColor` $\in$ {`Blond`, `Dark`} in the training data. The rarest group which is blond males represents only 0.85 % of the training data (1387 out of 162k examples). We train a ResNet-50 model [35] on this task. Tab. 3 summarizes the results and compares the performance of several methods. A model with vanilla cross-entropy (ERM) appears to generalize well on average but fails to generalize to the rarest group (blond males) which can be considered as "weakly" out-of-distribution (OOD). Our proposed SD improves the performance more than twofold. It should be highlighted that for this task, we use a variant of SD in which, $\frac{\lambda}{2}||\hat{y} - \gamma||_2^2$ is added to the original cross-entropy loss. The hyper-parameters λ and γ are tuned separately for each class (a total of four hyper-parameters). This variant of SD does provably decouple the dynamics too but appears to perform better than the original SD in Eq. 17 in this task.

Other proposed methods presented in Tab. 3 also show significant improvements on the performance of the worst group accuracy. The recently proposed "Learning from failure" (LfF) [69] achieves comparable results to SD, but it requires simultaneous training of two networks. Group DRO [86] is another successful method for this task. However, unlike SD, Group DRO requires explicit

information about the spuriously correlated attributes. In most practical tasks, information about the spurious correlations is not provided and, dependence on the spurious correlation goes unrecognized.[1]

5 Related Work and Discussion

Here, we discuss the related work. Due to space constraints, further discussions are presented SM. A. We especially highlight that potential negative outcomes of GS and SD are discussed in subsection *Gradient Starvation: friend or foe?"*.

On learning dynamics and Loss Choice. Several works including [87, 88, 1, 57] investigate the dynamics of deep linear networks trained with squared-error loss. Different decompositions of the learning process for neural networks have been used: [81, 99, 84, 100] study the learning in the Fourier domain and show that low-frequency functions are learned earlier than high-frequency ones. [87, 2, 29] provide closed-form equations for the dynamics of linear networks in terms of the principal components of the input covariance matrix. More recently, with the introduction of neural tangent kernel (NTK) [48, 60], a new line of research is to study the convergence properties of gradient descent [e.g. 7, 67, 23, 26, 6, 41, 30, 105, 10, 96]. Among them, [11, 101, 18] decompose the learning process along the principal components of the NTK, and [21] show that the corresponding eigenvalues determines the convergence rates along every component. The message is that the training process can be decomposed into independent learning dynamics along the orthogonal directions (modes) of the input covariance matrix. However, distinct learning dynamics arise with cross-entropy loss —an objective function widely used in practice— rather then squared-error loss. To our knowledge, our study is the first to study such feature learning discrepancies with cross-entropy loss, making our contribution an important addition to this body of work.

On Gradient Starvation and no free lunch theorem. The *no free lunch* theorem [90, 98] states that "learning is impossible without making assumptions about training and test distributions". Perhaps, the most commonly used assumption of machine learning is the i.i.d. assumption [95], which assumes that training and test data are identically distributed. However, in general, this assumption might not hold, and in many practical applications, there are predictive features in the training set that do not generalize to the test set. A natural question that arises is *how to favor generalizable features over spurious features?* The most common approaches include *data augmentation, controlling the inductive biases, using regularizations,* and more recently *training using multiple environments*.

Here, we would like to elaborate on an interesting thought experiment of [77]: Suppose a neural network is provided with a chess book containing examples of chess games with the best movements indicated by a red arrow. The network can take two approaches: 1) learn how to play chess, or 2) learn just the red arrows. Either of these solutions results in zero training loss on the games in the book while only the former is generalizable to new games. With no external knowledge, the network typically learns the simpler solution.

Recent work aims to leverage the invariance principle across several environments to improve robust learning. This is akin to present several chess books to a network, each with markings indicating the best moves for different sets of games. In several studies [8, 56, 77, 3], methods are developed to aggregate information from multiple training environments in a way that favors the generalizable / domain-agnostic / invariant solution. We argue that even with having access to **only one** training environment, there is useful information in the training set that fails to be discovered due to Gradient Starvation. The information on how to actually play chess is already available in any of the chess books. Still, as soon as the network learns the red arrows, the network has no incentive for further learning. Therefore, *learning the red arrows is not an issue per se, but not learning to play chess is.*

6 Conclusion

In this paper, we formalized Gradient Starvation (GS) as a phenomenon that emerges when training with cross-entropy loss in neural networks. By analyzing the dynamical system corresponding to the learning process in a dual space, we showed that GS could slow down the learning of certain features, even if they are present in the training set. We derived spectral decoupling (SD) regularization as a possible remedy to GS.

[1] Recall that it took 3 years for the psychologist, Oskar Pfungst, to realize that Clever Hans was not capable of doing any arithmetic.

References

[1] Madhu S Advani and Andrew M Saxe. High-dimensional dynamics of generalization error in neural networks. *arXiv preprint arXiv:1710.03667*, 2017.

[2] Madhu S Advani, Andrew M Saxe, and Haim Sompolinsky. High-dimensional dynamics of generalization error in neural networks. *Neural Networks*, 2020.

[3] Kartik Ahuja, Karthikeyan Shanmugam, and Amit Dhurandhar. Linear regression games: Convergence guarantees to approximate out-of-distribution solutions. *arXiv preprint arXiv:2010.15234*, 2020.

[4] Kartik Ahuja, Karthikeyan Shanmugam, Kush Varshney, and Amit Dhurandhar. Invariant risk minimization games. *arXiv preprint arXiv:2002.04692*, 2020.

[5] Naveed Akhtar and Ajmal Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access*, 6:14410–14430, 2018.

[6] Zeyuan Allen-Zhu, Yuanzhi Li, and Yingyu Liang. Learning and generalization in overparameterized neural networks, going beyond two layers. In *Advances in neural information processing systems*, pages 6158–6169, 2019.

[7] Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, pages 242–252. PMLR, 2019.

[8] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[9] Sanjeev Arora, Nadav Cohen, Wei Hu, and Yuping Luo. Implicit regularization in deep matrix factorization. In *Advances in Neural Information Processing Systems*, pages 7413–7424, 2019.

[10] Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Russ R Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. In *Advances in Neural Information Processing Systems*, pages 8139–8148, 2019.

[11] Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. *arXiv preprint arXiv:1901.08584*, 2019.

[12] Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. *arXiv preprint arXiv:1706.05394*, 2017.

[13] Nicholas Baker, Hongjing Lu, Gennady Erlikhman, and Philip J Kellman. Deep convolutional networks do not classify based on global object shape. *PLoS computational biology*, 14(12):e1006613, 2018.

[14] Pierre Baldi and Kurt Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural networks*, 2(1):53–58, 1989.

[15] Aristide Baratin, Thomas George, César Laurent, R Devon Hjelm, Guillaume Lajoie, Pascal Vincent, and Simon Lacoste-Julien. Implicit regularization in deep learning: A view from function space. *arXiv preprint arXiv:2008.00938*, 2020.

[16] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 456–473, 2018.

[17] Yonatan Belinkov and Yonatan Bisk. Synthetic and natural noise both break neural machine translation. *arXiv preprint arXiv:1711.02173*, 2017.

[18] Alberto Bietti and Julien Mairal. On the inductive bias of neural tangent kernels. In *Advances in Neural Information Processing Systems*, pages 12893–12904, 2019.

[19] Wieland Brendel and Matthias Bethge. Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. *arXiv preprint arXiv:1904.00760*, 2019.

[20] 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.

[21] Yuan Cao, Zhiying Fang, Yue Wu, Ding-Xuan Zhou, and Quanquan Gu. Towards understanding the spectral bias of deep learning. *arXiv preprint arXiv:1912.01198*, 2019.

[22] Yuan Cao and Quanquan Gu. Generalization bounds of stochastic gradient descent for wide and deep neural networks. In *Advances in Neural Information Processing Systems*, pages 10836–10846, 2019.

[23] Lenaic Chizat and Francis Bach. A note on lazy training in supervised differentiable programming. *arXiv preprint arXiv:1812.07956*, 1, 2018.

[24] Remi Tachet des Combes, Mohammad Pezeshki, Samira Shabanian, Aaron Courville, and Yoshua Bengio. On the learning dynamics of deep neural networks. *arXiv preprint arXiv:1809.06848*, 2018.

[25] Simon S Du, Wei Hu, and Jason D Lee. Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced. In *Advances in Neural Information Processing Systems*, pages 384–395, 2018.

[26] Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. *arXiv preprint arXiv:1810.02054*, 2018.

[27] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *arXiv preprint arXiv:2004.07780*, 2020.

[28] Thomas George. Nngeometry: Easy and fast fisher information matrices and neural tangent kernels in pytorch, 2020.

[29] Gauthier Gidel, Francis Bach, and Simon Lacoste-Julien. Implicit regularization of discrete gradient dynamics in linear neural networks. In *Advances in Neural Information Processing Systems*, pages 3202–3211, 2019.

[30] Sebastian Goldt, Madhu Advani, Andrew M Saxe, Florent Krzakala, and Lenka Zdeborová. Dynamics of stochastic gradient descent for two-layer neural networks in the teacher-student setup. In *Advances in Neural Information Processing Systems*, pages 6981–6991, 2019.

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

[32] 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.

[33] Suriya Gunasekar, Blake E Woodworth, Srinadh Bhojanapalli, Behnam Neyshabur, and Nati Srebro. Implicit regularization in matrix factorization. In *Advances in Neural Information Processing Systems*, pages 6151–6159, 2017.

[34] Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R Bowman, and Noah A Smith. Annotation artifacts in natural language inference data. *arXiv preprint arXiv:1803.02324*, 2018.

[35] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[36] Christina Heinze-Deml and Nicolai Meinshausen. Conditional variance penalties and domain shift robustness. *arXiv preprint arXiv:1710.11469*, 2017.

[37] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

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

[39] Katherine L Hermann and Andrew K Lampinen. What shapes feature representations? exploring datasets, architectures, and training. *arXiv preprint arXiv:2006.12433*, 2020.

[40] Tom M Heskes and Bert Kappen. On-line learning processes in artificial neural networks. In *North-Holland Mathematical Library*, volume 51, pages 199–233. Elsevier, 1993.

[41] Jiaoyang Huang and Horng-Tzer Yau. Dynamics of deep neural networks and neural tangent hierarchy. *arXiv preprint arXiv:1909.08156*, 2019.

[42] Like Hui and Mikhail Belkin. Evaluation of neural architectures trained with square loss vs cross-entropy in classification tasks. *arXiv preprint arXiv:2006.07322*, 2020.

[43] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. *arXiv preprint arXiv:1804.08598*, 2018.

[44] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems*, pages 125–136, 2019.

[45] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[46] Tommi S Jaakkola and David Haussler. Probabilistic kernel regression models. In *AISTATS*, 1999.

[47] Jörn-Henrik Jacobsen, Jens Behrmann, Richard Zemel, and Matthias Bethge. Excessive invariance causes adversarial vulnerability. *arXiv preprint arXiv:1811.00401*, 2018.

[48] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pages 8571–8580, 2018.

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

[50] Jason Jo and Yoshua Bengio. Measuring the tendency of cnns to learn surface statistical regularities. *arXiv preprint arXiv:1711.11561*, 2017.

[51] Alexia Jolicoeur-Martineau and Ioannis Mitliagkas. Connections between support vector machines, wasserstein distance and gradient-penalty gans. *arXiv preprint arXiv:1910.06922*, 2019.

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

[53] Simon Kornblith, Honglak Lee, Ting Chen, and Mohammad Norouzi. What's in a loss function for image classification? *arXiv preprint arXiv:2010.16402*, 2020.

[54] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[55] Anders Krogh and John A Hertz. A simple weight decay can improve generalization. In *Advances in neural information processing systems*, pages 950–957, 1992.

[56] David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (rex). *arXiv preprint arXiv:2003.00688*, 2020.

[57] Andrew K Lampinen and Surya Ganguli. An analytic theory of generalization dynamics and transfer learning in deep linear networks. *arXiv preprint arXiv:1809.10374*, 2018.

[58] Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Unmasking clever hans predictors and assessing what machines really learn. *Nature communications*, 10(1):1–8, 2019.

[59] Yann Le Cun, Ido Kanter, and Sara A Solla. Eigenvalues of covariance matrices: Application to neural-network learning. *Physical Review Letters*, 66(18):2396, 1991.

[60] Jaehoon Lee, Lechao Xiao, Samuel Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. In *Advances in neural information processing systems*, pages 8570–8581, 2019.

[61] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*, pages 7167–7177, 2018.

[62] 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.

[63] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[64] Cong Ma, Kaizheng Wang, Yuejie Chi, and Yuxin Chen. Implicit regularization in nonconvex statistical estimation: Gradient descent converges linearly for phase retrieval and matrix completion. In *International Conference on Machine Learning*, pages 3345–3354. PMLR, 2018.

[65] 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.

[66] R Thomas McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. *arXiv preprint arXiv:1902.01007*, 2019.

[67] Song Mei and Andrea Montanari. The generalization error of random features regression: Precise asymptotics and double descent curve. *arXiv preprint arXiv:1908.05355*, 2019.

[68] 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.

[69] Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. Learning from failure: Training debiased classifier from biased classifier. *arXiv preprint arXiv:2007.02561*, 2020.

[70] Kamil Nar, Orhan Ocal, S Shankar Sastry, and Kannan Ramchandran. Cross-entropy loss and low-rank features have responsibility for adversarial examples. *arXiv preprint arXiv:1901.08360*, 2019.

[71] Kamil Nar and S Shankar Sastry. Persistency of excitation for robustness of neural networks. *arXiv preprint arXiv:1911.01043*, 2019.

[72] Behnam Neyshabur, Ryota Tomioka, Ruslan Salakhutdinov, and Nathan Srebro. Geometry of optimization and implicit regularization in deep learning. *arXiv preprint arXiv:1705.03071*, 2017.

[73] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv preprint arXiv:1412.6614*, 2014.

[74] Timothy Niven and Hung-Yu Kao. Probing neural network comprehension of natural language arguments. *arXiv preprint arXiv:1907.07355*, 2019.

[75] Luke Oakden-Rayner, Jared Dunnmon, Gustavo Carneiro, and Christopher Ré. Hidden stratification causes clinically meaningful failures in machine learning for medical imaging. In *Proceedings of the ACM Conference on Health, Inference, and Learning*, pages 151–159, 2020.

[76] Samet Oymak, Zalan Fabian, Mingchen Li, and Mahdi Soltanolkotabi. Generalization guarantees for neural networks via harnessing the low-rank structure of the jacobian. *arXiv preprint arXiv:1906.05392*, 2019.

[77] Giambattista Parascandolo, Alexander Neitz, Antonio Orvieto, Luigi Gresele, and Bernhard Schölkopf. Learning explanations that are hard to vary. *arXiv preprint arXiv:2009.00329*, 2020.

[78] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[79] Oskar Pfungst. *Clever Hans:(the horse of Mr. Von Osten.) a contribution to experimental animal and human psychology*. Holt, Rinehart and Winston, 1911.

[80] Tomaso Poggio, Kenji Kawaguchi, Qianli Liao, Brando Miranda, Lorenzo Rosasco, Xavier Boix, Jack Hidary, and Hrushikesh Mhaskar. Theory of deep learning iii: explaining the non-overfitting puzzle. *arXiv preprint arXiv:1801.00173*, 2017.

[81] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International Conference on Machine Learning*, pages 5301–5310. PMLR, 2019.

[82] Jonas Rauber, Wieland Brendel, and Matthias Bethge. Foolbox: A python toolbox to benchmark the robustness of machine learning models. *arXiv preprint arXiv:1707.04131*, 2017.

[83] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[84] Basri Ronen, David Jacobs, Yoni Kasten, and Shira Kritchman. The convergence rate of neural networks for learned functions of different frequencies. In *Advances in Neural Information Processing Systems*, pages 4761–4771, 2019.

[85] Amir Rosenfeld, Richard Zemel, and John K Tsotsos. The elephant in the room. *arXiv preprint arXiv:1808.03305*, 2018.

[86] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

[87] Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

[88] Andrew M Saxe, James L McClelland, and Surya Ganguli. A mathematical theory of semantic development in deep neural networks. *Proceedings of the National Academy of Sciences*, 116(23):11537–11546, 2019.

[89] Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. *arXiv preprint arXiv:2006.07710*, 2020.

[90] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[91] 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.

[92] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[93] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

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

[95] V Vapnik and Vlamimir Vapnik. Statistical learning theory wiley. *New York*, pages 156–160, 1998.

[96] Santosh Vempala and John Wilmes. Gradient descent for one-hidden-layer neural networks: Polynomial convergence and sq lower bounds. In *Conference on Learning Theory*, pages 3115–3117, 2019.

[97] Haohan Wang, Zexue He, Zachary C Lipton, and Eric P Xing. Learning robust representations by projecting superficial statistics out. *arXiv preprint arXiv:1903.06256*, 2019.

[98] David H Wolpert. The lack of a priori distinctions between learning algorithms. *Neural computation*, 8(7):1341–1390, 1996.

[99] Zhi-Qin John Xu, Yaoyu Zhang, Tao Luo, Yanyang Xiao, and Zheng Ma. Frequency principle: Fourier analysis sheds light on deep neural networks. *arXiv preprint arXiv:1901.06523*, 2019.

[100] Zhi-Qin John Xu, Yaoyu Zhang, and Yanyang Xiao. Training behavior of deep neural network in frequency domain. In *International Conference on Neural Information Processing*, pages 264–274. Springer, 2019.

[101] Greg Yang and Hadi Salman. A fine-grained spectral perspective on neural networks. *arXiv preprint arXiv:1907.10599*, 2019.

[102] John R Zech, Marcus A Badgeley, Manway Liu, Anthony B Costa, Joseph J Titano, and Eric K Oermann. Confounding variables can degrade generalization performance of radiological deep learning models. *arXiv preprint arXiv:1807.00431*, 2018.

[103] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

[104] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. *arXiv preprint arXiv:1707.09457*, 2017.

[105] Difan Zou, Yuan Cao, Dongruo Zhou, and Quanquan Gu. Gradient descent optimizes over-parameterized deep relu networks. *Machine Learning*, 109(3):467–492, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See Section 1 for the list of our contributions.
   (b) Did you describe the limitations of your work? [Yes] In Section 4.3, we describe how similar to all the existing methods on OOD tasks, SD assumes access to the test environment for hyperparameter tuning. This is not a valid assumption in general, and hence the results should be only interpreted as a probe that shows that SD could provide an important level of control over what features are learned.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] This work is an effort to improve out of distribution generalization which to the best of our knowledge, mostly has positive impact and improves many tasks such as medical imaging and fairness. However the main text explicitly refers to a discussion section in Section SM A ("Gradient Starvation: friend or foe?") that considers potential negative impacts.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] .
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] We make three key assumption that are described in Section 3 and also Assumption 1.
   (b) Did you include complete proofs of all theoretical results? [Yes] Complete proofs of all our results are provided in the supplementary materials.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We provide a URL to an anonymous GitHub repository where we include all code and experiments details. See Section 1 for the link.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We have included all the experiments details in the GitHub repository further details are also available in the supplementary materials B.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Wherever applicable, we report our results based on 10 random runs and report the mean and the standard deviation in the Tables.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] The details of our experiments are included in the supplementary materials B. We use a GPU cluster to run the experiments.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes] In the GitHub repository, for each experiment, we have provided a folder which contains the relevant assets and its license.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We have included the source code of our experiments as a URL.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] Our data does not include personally identifiable information or offensive content.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]