Gradient Starvation: A Learning Proclivity in Neural Networks

Mohammad Pezeshki[1,2] **Sékou-Oumar Kaba**[1,3] **Yoshua Bengio**[1,2]
Aaron Courville[1,2] **Doina Precup**[1,3,4] **Guillaume Lajoie**[1,2]
[1]Mila [2]Université de Montréal [3]McGill University [4]Google DeepMind
corresponding authors:{pezeshki, guillaume.lajoie}@mila.quebec

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 [81]. 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. 85, 109, 54, 39, 17, 14, 37, 51, 107, 76, 48, 19, 61, 77]. 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, 88, 40, 68, 30].

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 [97, 70, 22, 73, 51, 76, 100, 92, 83, 105, 42, 79, 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 [30, 68, 77, 41, 63, 64, 9]. 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 [26], is a leading cause for this *feature imbalance* in neural networks trained with cross-entropy, and propose a simple approach to mitigate it.

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

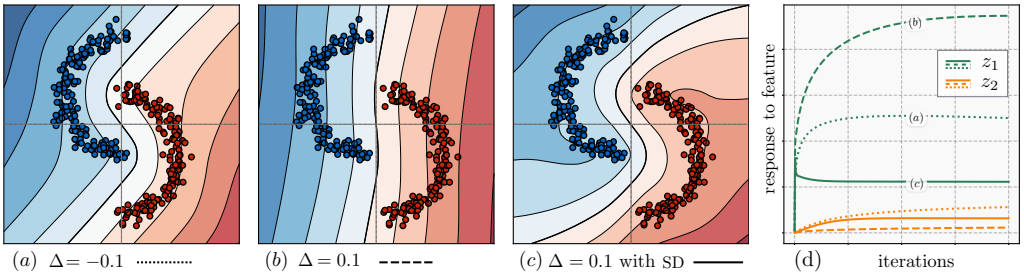

Figure 1: Diagram illustrating the effect of gradient starvation in a simple 2-D classification task. **(a)** Data is not linearly separable and the learned decision boundary is curved. **(b)** Data is linearly separable by a small margin ($\Delta = 0.1$). This small margin allows the network to discriminate 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 App. B.

Here we summarize our contributions:

- We provide a theoretical framework to study the learning dynamics of linearized neural networks trained with cross-entropy loss in a dual space.
- 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 [58], dropout [95], batch normalization [49], as well as changing the optimization algorithm to Adam [56] or changing the network architecture or the coordinate system, do not encourage the network to learn a curved decision boundary. (See App. 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.

2.1 Consequences of Gradient Starvation

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 [28] 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 [52], 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 [23],

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

In the wide-width regime, the NTRF changes very little during training [62], and the output of the neural network can be approximated by a first order Taylor expansion around the initialization parameters $\boldsymbol{\theta}_0$. Setting $\boldsymbol{\Phi}_0 \equiv \boldsymbol{\Phi}(\mathbf{X}, \boldsymbol{\theta}_0)$ 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

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

Dominant directions in the feature space as well as the parameter space are given by principal components of the NTRF matrix $\boldsymbol{\Phi}_0$, which are the same as those of the NTK Gram matrix [106]. We therefore introduce the following definition.

Definition 1 (Features and Responses)**.** *Consider the singular value decomposition (SVD) of the matrix* $\mathbf{Y}\boldsymbol{\Phi}_0 = \mathbf{U}\mathbf{S}\mathbf{V}^T$*, 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* z_j *where,*

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

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)**.** *Recall the the model prescribed by Eq. 3. Let z_j^* denote the model's response to feature j at training optimum $\boldsymbol{\theta}^*$*[1]*. 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 [26]. 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 [50], 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, $\boldsymbol{\alpha} \in (0,1)^n$ is a variational parameter defined for each training example, and $\odot$ denotes the element-wise vector product. 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 [50]). Since the neural network's output is approximated by a linear function of $\boldsymbol{\theta}$, the minimization can be performed analytically with an critical 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,

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

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),

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

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.

[1] Training optimum refers to the solution to $\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}) = 0$.

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,

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

See App. A for further discussions on the distinction between "feature space" and "parameter space". 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.

Lemma 1. *Assume* $\mathbf{U}$ *is a perturbed identity matrix (a special case of a permutation matrix) in which the off-diagonal elements are proportional to a small parameter* $\delta > 0$. *Then, the fixed point of the dynamical system in Eq. 10 can be approximated by,*

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

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 critical 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. For further intuition, we provide a simple experiment, summarized in Fig. 5, where directly visualizes the primal vs. the dual dynamics as well as the effect of the proposed spectral decoupling method.

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 including the scheme for hyper-parameter tuning, see App. 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 App. 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) [57] 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 [72], we use "the norm of input-disturbance required to cross the decision boundary" as a proxy for the margin. The disturbance on the input is computed by projected gradient descent (PGD) [84], a well-known adversarial attack.

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 [72]). SD consistently achieves a better OOD performance.

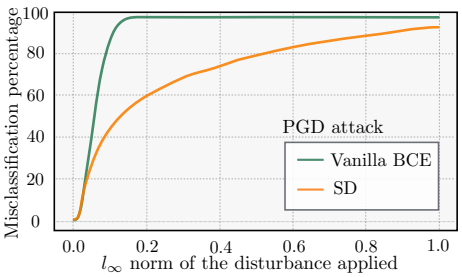

Figure 2: The plot shows the cumulative distribution function (CDF) of the margin for the CIFAR-2 binary classification. SD appears to improve the margin considerably.

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 percentange 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. See Section 5 for a discussion on why cross-entropy results in a poor margin while being considered a max-margin classifier in the literature [94].

4.3 Colored MNIST with color bias

We conduct experiments on the Colored MNIST Dataset, proposed in [9]. 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) 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 [59]	71.5 % (±1.0)	68.7 % (±0.9)
IRM [9]	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) [9] 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. Risk Extrapolation (REx) [59] is a related training method that encourages learning invariant representations. Similar to IRM, it requires access to multiple training environments in order to quantify the concept of "invariance".

SD achieves an accuracy of 68.4 %. Its performance is remarkable because unlike IRM and REx, 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

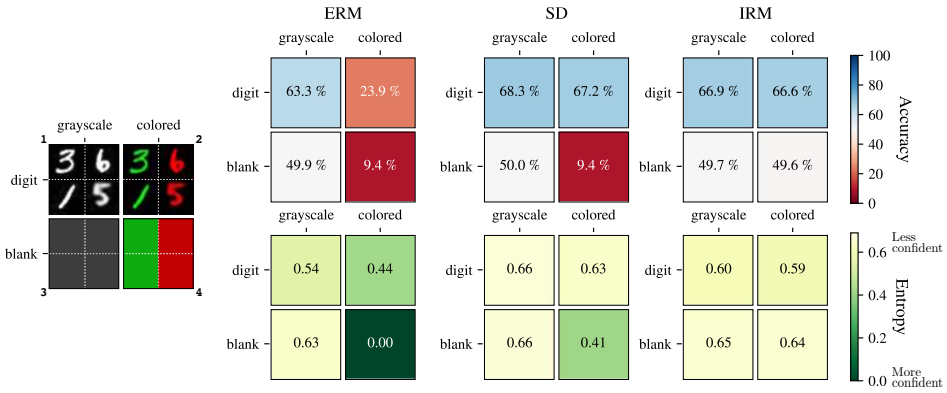

Figure 3: Diagram comparing ERM, SD, and IRM on four different test environments on which we evaluate a pre-trained model. Top and bottom rows show the accuracy and the entropy (inverse of confidence), respectively. **Analysis:** Compare three values of 9.4 %, 9.4 %, and 49.6 % : Both ERM and SD have 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.

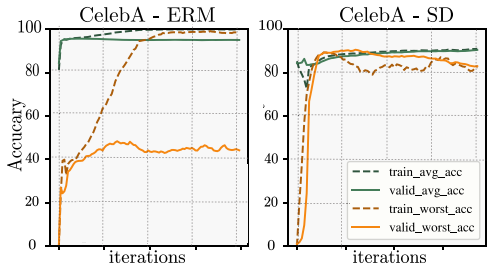

Figure 4: CelebA: `blond` vs `dark` hair classification. The `HairColor` and the `Gender` are spuriously correlated which leads to poor OOD performance with ERM, however SD significantly improves performance. ERM's worst group accuracy is significantly lower than SD.

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 [89] requires explicit information about the spurious correlation. LfF [71] requires simultaneous training of two networks.

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.

4.4 CelebA with gender bias

The CelebA dataset [65] contains 162k celebrity faces with binary attributes associated with each image. Following the setup of [89], 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 [38] 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) [71] achieves comparable results to SD, but it requires simultaneous training of two networks. Group DRO [89] 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.[2]

5 Related Work and Discussion

Here, we discuss the related work. Due to space constraints, further discussions are in App. A.

On learning dynamics and Loss Choice. Several works including [90, 91, 1, 60] investigate the dynamics of deep linear networks trained with squared-error loss. Different decompositions of the learning process for neural networks have been used: [83, 104, 87, 105] study the learning in the Fourier domain and show that low-frequency functions are learned earlier than high-frequency ones. [90, 2, 32] 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) [52, 62], a new line of research is to study the convergence properties of gradient descent [e.g. 8, 69, 25, 29, 7, 44, 33, 110, 11, 99]. Among them, [12, 106, 18, 22] decompose the learning process along the principal components of the NTK. The message in these works is that the training process can be decomposed into independent learning dynamics along the orthogonal directions.

Most of the studies mentioned above focus on the particular squared-error loss. For a linearized network, the squared-error loss results in linear learning dynamics, which often admit an analytical solution. However, the de-facto loss function for many of the practical applications of neural networks is the cross-entropy. Using the cross-entropy as the loss function leads to significantly more complicated and non-linear dynamics, even for a linear neural network. In this work, our focus was the cross-entropy loss.

On reliance upon spurious correlations and robustness. In the context of robustness in neural networks, state-of-the-art neural networks appear to naturally focus on low-level superficial correlations rather than more abstract and robustly informative features of interest (e.g. [30]). As we argue in this work, Gradient Starvation is likely an important factor contributing to this phenomenon and can result in adversarial vulnerability. There is a rich research literature on adversarial attacks and neural networks' vulnerability [96, 34, 48, 67, 5, 47]. Interestingly, [73], [72] and [51] draw a similar conclusion and argue that "an insufficiency of the cross-entropy loss" causes excessive invariances to predictive features. Perhaps [92] is the closest to our work in which authors study the simplicity bias (SB) in stochastic gradient descent. They demonstrate that neural networks exhibit extreme bias that could lead to adversarial vulnerability.

On implicit bias. Despite being highly-overparameterized, modern neural networks seem to generalize very well [108]. Modern neural networks generalize surprisingly well in numerous machine tasks. This is despite the fact that neural networks typically contain orders of magnitude more parameters than the number of examples in a training set and have sufficient capacity to fit a totally randomized dataset perfectly [108]. The widespread explanation is that the gradient descent has a form of implicit bias towards learning simpler functions that generalize better according to Occam's razor. Our exposition of GS reinforces this explanation. In essence, when training and test data points are drawn from the same distribution, the top salient features are predictive in both sets. We conjecture that in such a scenario, by not learning the less salient features, GS naturally protects the network from overfitting.

The same phenomenon is referred to as *implicit bias*, *implicit regularization*, *simplicity bias* and *spectral bias* in several works [83, 75, 36, 74, 70, 53, 94, 10, 13, 35, 82, 66].

As an active line of research, numerous studies have provided different explanations for this phenomenon. For example, [70] justifies the implicit bias of neural networks by showing that stochastic gradient descent learns simpler functions first. [15, 78] suggests that a form of implicit regularization is induced by an alignment between NTK's principal components and only a few task-relevant

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

directions. Several other works such as [20, 35, 94, 25] recognize the convergence of gradient descent to maximum-margin solution as the essential factor for the generalizability of neural networks. It should be stressed that these work refer to the margin in the hidden space and not in the input space as pointed out in [55]. Indeed, as observed in our experiments, the maximum-margin classifier in the hidden space can be achieved at the expense of a small margin in the input space.

On Gradient Starvation and no free lunch theorem. The *no free lunch* theorem [93, 102] 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 [98], 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 [79]: 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 [9, 59, 79, 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.*

Gradient Starvation: friend or foe? Here, we would like to remind the reader that GS can have both adverse and beneficial consequences. If the learned features are sufficient to generalize to the test data, gradient starvation can be viewed as an implicit regularizer. Otherwise, Gradient Starvation could have an unfavorable effect, which we observe empirically when some predictive features fail to be learned. A better understanding and control of Gradient Starvation and its impact on generalization offers promising avenues to address this issue with minimal assumptions. Indeed, our Spectral Decoupling method requires an assumption about feature imbalance but not to pinpoint them exactly, relying on modulated learning dynamics to achieve balance.

GS social impact Modern neural networks are being deployed extensively in numerous machine learning tasks. Our models are used in critical applications such as autonomous driving, medical prediction, and even justice system where human lives are at stake. However, neural networks appear to base their predictions on superficial biases in the dataset. Unfortunately, biases in datasets could be neglected and pose negative impacts on our society. In fact, our Celeb-A experiment is an example of the existence of such a bias in the data. As shown in the paper, the gender-specific bias could lead to a superficial high performance and is indeed very hard to detect. Our analysis, although mostly on the theory side, could pave the path for researchers to build machine learning systems that are robust to biases and helps towards fairness in our predictions.

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.

Acknowledgments and Disclosure of Funding

The authors are grateful to Samsung Electronics Co., Ldt., CIFAR, and IVADO for their funding and Calcul Québec and Compute Canada for providing us with the computing resources. We would further like to acknowledge the significance of discussions and supports from Reyhane Askari Hemmat and Faruk Ahmed. MP would like to thank Aristide Baratin, Kostiantyn Lapchevskyi, Seyed Mohammad Mehdi Ahmadpanah, Milad Aghajohari, Kartik Ahuja, Shagun Sodhani, and Emmanuel Bengio for their invaluable help.

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] Adelin Albert and John A Anderson. On the existence of maximum likelihood estimates in logistic regression models. *Biometrika*, 71(1):1–10, 1984.

[7] 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.

[8] 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.

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

[10] 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.

[11] 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 8141–8150, 2019.

[12] 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.

[13] 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.

[14] 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.

[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] Christopher JC Burges and David J Crisp. Uniqueness of the svm solution. *Advances in neural information processing systems*, 12:223–229, 2000.

[22] 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.

[23] 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.

[24] Zixiang Chen, Yuan Cao, Quanquan Gu, and Tong Zhang. A generalized neural tangent kernel analysis for two-layer neural networks. *arXiv preprint arXiv:2002.04026*, 2020.

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

[26] 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.

[27] Thomas M Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE transactions on electronic computers*, 0(3):326–334, 1965.

[28] 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.

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

[30] 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.

[31] Thomas George. Nngeometry: Easy and fast fisher information matrices and neural tangent kernels in pytorch. *0*, 2020.

[32] 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.

[33] 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.

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

[35] 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.

[36] 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.

[37] 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.

[38] 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.

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

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

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

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

[43] Cho-Jui Hsieh, Kai-Wei Chang, Chih-Jen Lin, S Sathiya Keerthi, and Sellamanickam Sundararajan. A dual coordinate descent method for large-scale linear svm. In *Proceedings of the 25th international conference on Machine learning*, pages 408–415, 2008.

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

[45] Kaixuan Huang, Yuqing Wang, Molei Tao, and Tuo Zhao. Why do deep residual networks generalize better than deep feedforward networks?—a neural tangent kernel perspective. *Advances in Neural Information Processing Systems*, 33, 2020.

[46] 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.

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

[48] 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.

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

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

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

[52] 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.

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

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

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

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

[57] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *0*, 2009.

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

[59] 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.

[60] 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.

[61] 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.

[62] 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.

[63] 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.

[64] 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.

[65] 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.

[66] 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.

[67] 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.

[68] 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.

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

[70] 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.

[71] 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.

[72] 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.

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

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

[75] 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.

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

[77] 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.

[78] 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.

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

[80] 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. *openreview id=BJJsrmfCZ*, 2017.

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

[82] 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.

[83] 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.

[84] 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.

[85] 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.

[86] Michael Roberts. Machine learning for covid-19 diagnosis: Promising, but still too flawed, Mar 2021.

[87] 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.

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

[89] 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.

[90] 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.

[91] 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.

[92] 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.

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

[94] 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.

[95] 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.

[96] 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.

[97] 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.

[98] Vladimir Vapnik and Vlamimir Vapnik. Statistical learning theory wiley. *New York*, 1:624, 1998.

[99] 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.

[100] 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.

[101] Sifan Wang, Xinling Yu, and Paris Perdikaris. When and why pinns fail to train: A neural tangent kernel perspective. *Journal of Computational Physics*, page 110768, 2021.

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

[103] Xin Xu and Eibe Frank. Logistic regression and boosting for labeled bags of instances. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 272–281. Springer, 2004.

[104] 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.

[105] 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.

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

[107] 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.

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

[109] 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.

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

A Further discussions

On Primal (parameter space) vs. Dual (feature space) dynamics: Although the cross-entropy loss is convex, it does not admit an analytical solution, even in a simple logistic regression [103]. Importantly, it also does not have a finite solution when the data is linearly separable [6] (which is the case in high dimensions [27]). As such, our study is concerned with characterizing the solutions that the training algorithm converges to. A dual optimization approach enables us to describe these solutions in terms of contributions of the training examples [43]. While primal and dual dynamics are not guaranteed to match, the solution they converge to is guaranteed to match [21], and that is what our theory builds upon.

For further intuition, we provide a simple experiment in app C, directly visualizing the primal vs. the dual dynamics as well as the effect of the proposed spectral decoupling method.

The intuition behind Spectral Decoupling (SD): Consider a training datapoint x in the middle of the training process. Intuitively, the model has two options for decreasing the loss of this example:

1. Get more confident on a feature that has been learned already by other examples. or,
2. Learn a new feature.

SD, a simple L2 penalty on the output of the work, would favor (2) over (1). The reason is that (2) does not make the network over-confident on previously learned examples, while (1) results in over-confident predictions. Hence, SD encourages learning more features by penalizing confidence. Our principal novel contribution is to characterize this process formally and to theoretically and empirically demonstrate its effectiveness.

From another perspective, here we describe how one can arrive at Spectral Decoupling. From Thm. 2, we know that Gradient Starvation happens because of the coupling between features (equivalently alphas). We notice that in Eq. 9, if we get rid of S^2, then the alphas are decoupled. To get rid of S^2 , one can see that instead of $||\boldsymbol{\theta}||^2$ as the regularizer, we should have $||SV^T\boldsymbol{\theta}||^2$. Luckily, this is exactly equal to $||\hat{y}^2||$, since $\hat{y} = \Phi\boldsymbol{\theta} = UV^T\boldsymbol{\theta}$. We would like to highlight that $||SV^T\boldsymbol{\theta}||^2$ as the regularizer means that different directions are penalized according to their strength. It means that we suppress stronger directions more than others which would allow weaker directions to flourish.

Then why not use Squared-error loss for classification too? The biggest obstacle when using squared-error loss for classification is how to select the target. For example, in a cats vs. dogs classification task, not all cats have the same amount of "catty features". However, recent results favor using squared-error loss for classification and show that models trained with squared-error loss are more robust [46]. We conjecture that the improved robustness can be attributed to a lack of gradient starvation.

On using NTK: Theoretical analysis of neural networks in their general form is challenging and generally intractable. Neural Tangent Kernel (NTK) has been an important milestone that has simplified theoretical analysis significantly and provides some mechanistic explanations that are applicable in practice. Inevitably, it imposes a set of restrictions; mainly, NTK is only accurate in the limit of large width. Therefore, the common practice is to provide the theoretical analysis in simplified settings and validate the results empirically in more general cases (see, e.g. [45, 24, 101]). In this work, we build on the same established practices: Our theories analytically study an NTK linearized network; and we further validate our findings on several standard neural networks. In fact, in all of our experiments, learning is done in the regular "rich" (non-NTK) regime, and we verify that our proposed method, as identified analytically, mitigates learning limitations.

Future Directions: This work takes a step towards understanding the reliance of neural networks upon spurious correlations and shortcuts in the dataset. We believe identifying this reliance in sensitive applications is among the next steps for future research directions. That would have a pronounced real-world impact as neural networks have started to be used in many critical applications. As a recent example, we would like to point to an article by researchers at Cambridge [86] where they study more than 300 papers on detecting whether a patient has COVID or not given their CT Scans. According to the article, none of the papers were able to generalize from one hospital data to another since the models learn to latch on to hospital-specific features. An essential first step is to uncover such reliance and then to design methods such as our proposed spectral decoupling to mitigate the problem.

B Experimental Details

B.1 A Simple Experiment Summarizing the Theory

Here, we provide a simple experiment to study the difference between the primal and dual form dynamics. We also compare the learning dynamics in cases with and without Spectral Decoupling (SD).

Recall that primal dynamics arise from the following optimization,

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

while the dual dynamics are the result of another optimization,

$$\max_{\alpha_i} \left[H(\boldsymbol{\alpha}) - \frac{1}{2\lambda} \sum_{jq} \boldsymbol{\alpha}\mathbf{Y}\boldsymbol{\Phi}_0\boldsymbol{\Phi}_0^T\mathbf{Y}^T\boldsymbol{\alpha}^T \right].$$

Also recall that Spectral Decoupling suggests the following optimization,

$$\min_{\theta} \left(\mathbf{1} \cdot \log\left[1 + \exp\left(-\mathbf{Y}\hat{\mathbf{y}}\right)\right] + \frac{\lambda}{2}\|\hat{\mathbf{y}}\|^2 \right).$$

We conduct experiments on a simple toy classification with two datapoints for which the matrix $\mathbf{U}$ of Eq. 15 is defined as, $\mathbf{U} = \begin{pmatrix} 0.8 & -0.6 \\ 0.6 & 0.8 \end{pmatrix}$. The corresponding singular values $\mathbf{S} = [s_1, s_2 = 2]$ where $s_1 \in \{2, 3, 4, 5, 6\}$. According to Eq. 13, when $\mathbf{S} = [2, 2]$, the dynamics decouple while in other cases starvation occurs. Fig. 5 shows the corresponding features of z_1 and z_2. It is evident that by increasing the value of s_1, the value of z_1^* increases while z_2^* decreases (starves). Fig. 5 (left) also compares the difference between the primal and the dual dynamics. Note that although their dynamics are different, they both share the same fixed points. Fig. 5 (right) also shows that Spectral Decoupling (SD) indeed decouples the learning dynamics of z_1 and z_2 and hence increasing the corresponding singular value of one does not affect the other.

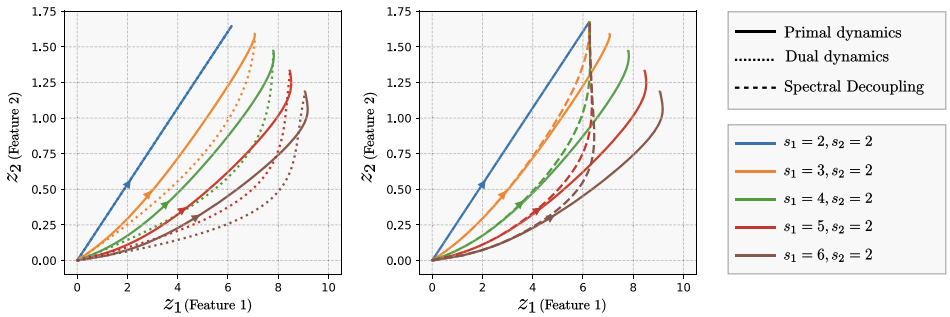

Figure 5: An illustration of the learning dynamics for a simple 2D classification task. x-axis and y-axis represent learning along features of z_1 and z_2, respectively. Each trajectory corresponds to a combination of the corresponding singular values of s_1 and s_2. It is evident that by increasing the value of s_1, the value of z_1^* increases while z_2^* decreases (starves). **(Left)** compares the difference between the primal and the dual dynamics. Note that although their dynamics are different, they both share the same fixed points. **(Right)** shows that Spectral Decoupling (SD) indeed decouples the learning dynamics of z_1 and z_2 and hence increasing the corresponding singular value of one does not affect the other.

B.2 Two-Moon Classification: Comparison with other regularization methods

We experiment the Two-moon classification example of the main paper with different regularization techniques. The small margin between the two classes allows the network to achieve a negligible loss by only learning to discriminate along the horizontal axis. However, both axes are relevant for the data distribution, and the only reason why the second dimension is not picked up is the fact that the

training data allows the learning to explain the labels with only one feature, overlooking the other. Fig. 6 reveals that common regularization strategies including Weight Decay, Dropout [95] and Batch Normalization [49] do not help achieving a larger margin classifier. Unless states otherwise, all the methods are trained with Full batch Gradient Descent with a learning rate of $1e-2$ and a momentum of 0.9 for $10k$ iterations.

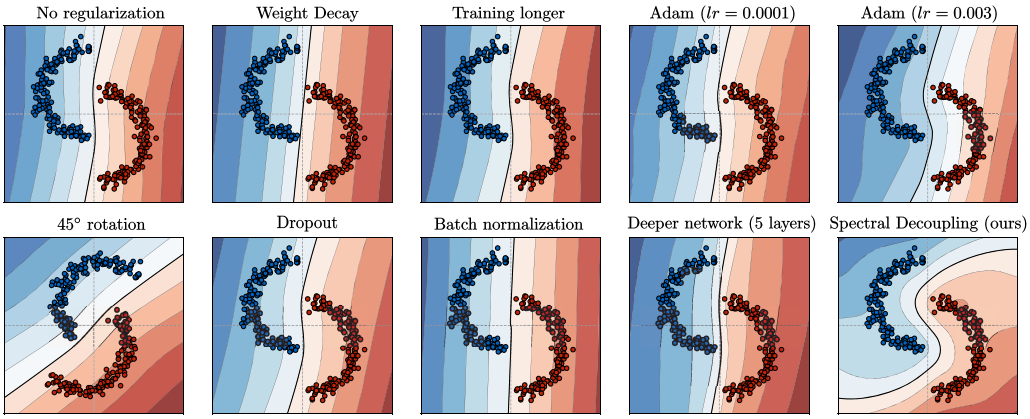

Figure 6: The effect of common regularization methods on a simple task of two-moon classification. It can be seen that common practices of deep learning seem not to help with learning a curved decision boundary. The acronym "lr" for the Adam optimizer refers to the learning rate. Shown decision boundaries are the average over 10 runs in which datapoints and the model initialization parameters are sampled randomly. Here, only the datapoints of one particular seed are plotted for visual clarity.

B.3 CIFAR classification

We use a four-layer convolutional network with ReLU non-linearity following the exact setup of [72]. Sweeping λ from 0 to its optimal value results in a smooth transition from green to orange. However, larger values of λ will hurt the IID test (zero perturbation) generalization. The value that we cross-validate on is the average of IID and OOD generalization performance.

B.4 Colored MNIST with color bias

For the Colored MNIST task, we aggregate all the examples from both training environments. Table. 4 reports the hyper-parameters used for each method.

Method	Layers	Dim	Weight decay	LR	Anneal steps	Penalty coef
ERM	2	300	0.0	1e-4	0/2000	n/a
SD	2	300	0.0	1e-4	450/2000	2e-5
IRM	2	390	0.00110794568	0.00048985365	190/500	91257.18613115903

Table 4: Hyper-parameters used for the Colored-MNIST experiment. Hyper-parameters of IRM are obtained from their released code. "Anneal steps" indicates the number of iterations done before applying the method.

More on Fig. 3. ERM captures the color feature and in the absence of any digit features (environment 4), the network's accuracy is low, as is expected because of reverse color-label match at testing. Moreover, the ERM network is very confident in this environment (confidence is inversely proportional to entropy). The SD network appears to capture the color feature too, with identical classification accuracy in environment 4, but much lower confidence which indicates the other features it expects to classify are absent. Consistent with this, in the case where both color and digit features are present (environment 2), SD achieves significantly better performance than ERM which is fooled by the flipped colors. This is again consistent with SD mitigating GS caused by the color feature onto the digit shape features. Meanwhile, IRM appears to not capture the color feature altogether. Specifically, when only the color is presented to a network trained with IRM, network predicts 50% accuracy

with low confidence meaning that IRM is indeed "invariant" to the color as its name suggests. We note that further theoretical justifications are required to fully understand the underlying mechanisms in learning with spurious correlations.

As a final remark, we 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.

The hyperparameter search has resulted in applying the SD at 450th step. We observe that 450th step is the step at which the traditional (in-distribution) overfitting occurs. This suggests that one might be able to tune hyperparameters without the need to monitor on the test set.

For all the experiments, we use PyTorch [80]. We also use NNGeometry [31] for computing NTK.

B.5 CelebA with gender bias: The experimental details

Figure 4 depicts the learning curves for this task with and without Spectral Decoupling. For the CelebA experiment, we follow the same setup as in [89] and use their released code. We use Adam optimizer for the Spectral Decoupling experiments with a learning rate of $1e-4$ and a batch size of 128. As mentioned in the main text, for this experiment, we use a different variant of Spectral Decoupling which also provably decouples the learning dynamics,

$$\min_{\theta} \; \mathbf{1} \cdot \left(\log\left[1 + \exp\left(-\mathbf{Y}\hat{\mathbf{y}}\right)\right] + \frac{\lambda}{2} \|\hat{\mathbf{y}} - \gamma\|^2 \right).$$

B.5.1 Hyper-parameters

We applied a hyper-parameter search on λ and γ for each of the classes separately. Therefore, a total of four hyper-parameters are found. For class zero, $\lambda_0 = 0.088$, $\gamma_0 = 0.44$ and for class one, $\lambda_1 = 0.012$, $\gamma_1 = 2.5$ are found to result in the best worst-group performance.

During the experiments, we found that for the CelebA dataset, classes are imbalanced: 10875 examples for class 0 and 1925 examples for class 1; meaning a ratio of 5.65. That is why we decided to penalize examples of each class separately with different coefficients. We also found that penalizing the outputs' distance to different values γ_0 and γ_1 helps the generalization. As stated in lines 842-844, the hyperparameter search results in the following values: 2.5 and 0.44.

B.6 Computational Resources

For the experiments and hyper-parameter search an approximate number of 800 GPU-hours has been used. GPUs used for the experiments are NVIDIA-V100 mostly on internal cluster and partly on public cloud clusters.

C Proofs of the Theories and Lemmas

C.1 Eq. 7 Legendre Transformation

Following [50], we derive the Legendre transformation of the Cross-Entropy (CE) loss function. Here, we reiterate this transformation as following,

Lemma 2 (CE's Legendre transformation, adapted from Eq. 46 of [50])**.** *For a variational parameter $\alpha \in [0, 1]$, the following linear lower bound holds for the cross-entropy loss function,*

$$\mathcal{L}(\omega) := \log\big(1 + e^{-\omega}\big) \geq H(\alpha) - \alpha\omega, \tag{19}$$

in which $\omega := y\hat{y}$ and $H(\alpha)$ is the Shannon's binary entropy. The equality holds for the critical value of $\alpha^* = -\nabla_\omega \mathcal{L}$, i.e., at the maximum of r.h.s. with respect to α.

Proof. The **Legendre** transformation converts a function $\mathcal{L}(\omega)$ to another function $g(\alpha)$ of conjugate variables α, $\mathcal{L}(\omega) \to g(\alpha)$. The idea is to find the expression of the tangent line to $\mathcal{L}(\omega)$ at ω_0 which is the first-order Taylor expansion of $\mathcal{L}(\omega)$,

$$t(\omega, \omega_0) = \mathcal{L}(\omega_0) + (\omega - \omega_0)\nabla_\omega \mathcal{L}|_{\omega=\omega_0}, \tag{20}$$

where $t(\omega, \omega_0)$ is the tangent line. According to the Legendre transformation, the function $\mathcal{L}(\omega)$ can be written as a function of the intercepts of tangent lines (where $\omega = 0$). Varying ω_0 along the x-axis provides us with a general equation, representing the intercept as a function of ω,

$$t(\omega = 0, \omega_0 = \omega) = \mathcal{L}(\omega) - \omega\nabla_\omega \mathcal{L}. \tag{21}$$

The cross-entropy loss function can be rewritten as a soft-plus function,

$$\mathcal{L}(\omega) = -\log \sigma(\omega) = \log\big(1 + e^{-\omega}\big), \tag{22}$$

in which $\omega := y\hat{y}$. Letting $\alpha := -\nabla_\omega \mathcal{L} = \sigma(-\omega)$ we have,

$$\omega = \log\left(\frac{1-\alpha}{\alpha}\right), \tag{23}$$

which allows us to re-write the expression for the intercepts as a function of α (denoted by $g(\alpha)$),

$$\begin{aligned} g(\alpha) &= \mathcal{L}(\omega) - \omega\nabla_\omega \mathcal{L} && (24)\\ &= \mathcal{L}(\omega) + \alpha\omega && (25)\\ &= -\alpha\log\alpha - (1-\alpha)\log(1-\alpha) && (26)\\ &= H(\alpha), && (27) \end{aligned}$$

where $H(\alpha)$ is the binary entropy function.

Now, since $\mathcal{L}$ is convex, a tangent line is always a lower bound and therefore at its maximum it touches the original function. Consequently, the original function can be recovered as follows,

$$\mathcal{L}(\omega) = \max_{0\le\alpha\le 1} \; H(\alpha) - \alpha\omega. \tag{28}$$

Note that the lower bound in Eq. 19 is now a linear function of $\omega := y\hat{y}$ but at the expense of an additional maximization over the variational parameter α. An illustration of the lower bound is depicted in Fig. 7. Also a comparison between the dual formulation of other common loss functions is provided in Table. 5.

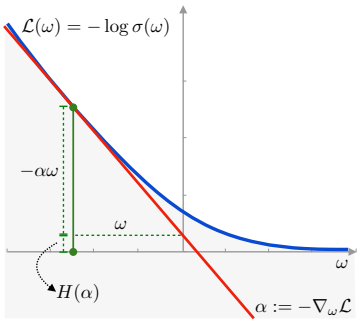

Figure 7: Diagram illustrating the Legendre transformation of the function $\mathcal{L}(\omega) = \log\big(1 + e^{-\omega}\big)$. The function is shown in blue, and the tangent line is shown in red. The tangent line is the lower bound of the function: $H(\alpha) - \alpha\omega \le \mathcal{L}(\omega)$.

Loss	**Primal form**	**Dual form**
Cross-Entropy	$\log(1+e^{-w})$	$\max\limits_{0<\alpha<1}\big[H(\alpha) - \alpha\omega\big]$
Hinge loss	$\max(0, 1-w)$	$\max\limits_{0\le\alpha\le 1}\big[\alpha\mathbf{1}^T - \alpha\omega\big]$
Squared error	$(1-\omega)^2$	$\max\limits_{\alpha}\big[-\frac{1}{2}\alpha^2 + \alpha - \alpha\omega\big]$

Table 5: Dual forms of other common different loss functions. The dual form of the Hinge loss is commonly used in Support Vector Machine (SVMs). For the ease of notation, we assume scalar ω and α.

□

C.1.1 Extension to Multi-Class

Building on Eq. 65-71 of [50], which derives the Legendre transform of multi-class cross-entropy, one can update Eq. 6 of the main paper to

$$-\sum_{c=1}^{C} y_c \log(\hat{y}_c) \geq H(\alpha) - \sum_{c=1}^{C} \alpha^c \hat{y}_c, \tag{29}$$

where $H(\alpha)$ is the entropy function, $C = \#$classes, and vectors of α^c are defined for each class. Then Eq. 8 of the paper is then updated to,

$$\big(H(\alpha) - \frac{1}{2\lambda}\sum_{c=1}^{C}(\delta_y - \alpha^c)\Phi\Phi^T(\delta_y - \alpha^c)^T\big). \tag{30}$$

With a change of variable $\alpha^c := \delta_y - \alpha^c$, the theory of SD should remain unchanged.

C.2 Eq. 8 Dual Dynamics

In Eq. 7, the order of min and max can be swapped as proved in Lemma 3 of [50], leading to,

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

The solution to the inner optimization is,

$$\boldsymbol{\theta}^{*T} = \frac{1}{\lambda}\boldsymbol{\alpha}\mathbf{Y}\boldsymbol{\Phi}_0,$$

which its substitution into Eq. 7 results in Eq. 8.

C.3 Eq. 9

Simply taking the derivative of Eq. 8 will result in Eq. 9. When we introduce continuous gradient ascent, we must define a learning rate parameter. This term is conceptually equivalent to the learning rate in SGD, but in this continuous setting, it has no influence on the fixed point.

C.4 Eq. 10 Approximate Dynamics

Approximating dynamics of Eq. 9 with a first order Taylor expansion around the origin of the second term, we obtain

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

Proof. Starting from the exact dynamics at Eq. 9,

$$\dot{\boldsymbol{\alpha}} = \eta \left(-\log \boldsymbol{\alpha} + \log\left(\mathbf{1} - \boldsymbol{\alpha}\right) - \frac{1}{\lambda}\boldsymbol{\alpha}\mathbf{U}\mathbf{S}^2\mathbf{U}^T \right), \tag{31}$$

we perform a first-order Taylor approximation of the second term at $\boldsymbol{\alpha} = \mathbf{0}$:

$$\log\left(\mathbf{1} - \boldsymbol{\alpha}\right) = -\boldsymbol{\alpha} + \mathcal{O}\left(\boldsymbol{\alpha}^2\right). \tag{32}$$

Replacing in Eq. 31, we obtain

$$\dot{\boldsymbol{\alpha}} \approx \eta \left(-\log \boldsymbol{\alpha} - \boldsymbol{\alpha} - \frac{1}{\lambda}\boldsymbol{\alpha}\mathbf{U}\mathbf{S}^2\mathbf{U}^T \right), \tag{33}$$

$$\dot{\boldsymbol{\alpha}} \approx \eta \left(-\log \boldsymbol{\alpha} - \boldsymbol{\alpha}\mathbf{U}\mathbf{I}\mathbf{U}^T - \frac{1}{\lambda}\boldsymbol{\alpha}\mathbf{U}\mathbf{S}^2\mathbf{U}^T \right), \tag{34}$$

$$\dot{\boldsymbol{\alpha}} \approx \eta \left(-\log \boldsymbol{\alpha} - \frac{1}{\lambda}\boldsymbol{\alpha}\mathbf{U}\left(\mathbf{S}^2 + \lambda\mathbf{I}\right)\mathbf{U}^T \right). \tag{35}$$

□

C.5 Thm. 1 Attractive Fixed-Points

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

Proof. We define

$$f_i(\alpha_j) = \eta \left(\log \frac{(1-\alpha_i)}{\alpha_i} - \frac{1}{\lambda} \sum_{jk} u_{ik} s_k^2 (u^T)_{kj} \alpha_j \right) \tag{36}$$

as the gradient function of the autonomous system Eq. 9.

We find the character of possible fixed points by linearization. We compute the jacobian of the gradient function evaluated at the fixed point.

$$J_{ik} = \left. \frac{\mathrm{d} f_i(\alpha_j)}{\mathrm{d}\alpha_k} \right|_{\alpha_k^*} \tag{37}$$

$$J_{ik} = \eta \left(-\delta_{ik} \left[\alpha_i^* (1 - \alpha_i^*) \right]^{-1} - \lambda^{-1} \sum_l u_{il} s_l^2 (u^T)_{lk} \right) \tag{38}$$

The fixed point is an attractor if the jacobian is a negative-definite matrix. The first term is negative-definite matrix while the second term is negative semi-definite matrix. Since the sum of a negative matrix and negative-semi definite matrix is negative-definite, this completes the proof. □

C.6 Eq. 11 Feature Response at Fixed-Point

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

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

Proof. The solution to the converged $\boldsymbol{\theta}^*$ at the Fixed-Point $\boldsymbol{\alpha}^*$ of Eq. 10 is,

$$\boldsymbol{\theta}^{*T} = \frac{1}{\lambda} \boldsymbol{\alpha}^* \mathbf{Y} \boldsymbol{\Phi}_0,$$

which by substitution into Eq. 4, Eq. 11 is derived.

□

C.7 Eq. 12 Uncoupled Case 1

If the matrix of singular values $\mathbf{S}^2$ is proportional to the identity, the fixed points of Eq. 10 are given by,

$$\alpha_i^* = \frac{\lambda \mathcal{W}(\lambda^{-1} s^2 + 1)}{s^2 + \lambda}, \qquad z_j^* = \frac{s^2 \mathcal{W}(\lambda^{-1} s^2 + 1)}{s^2 + \lambda} \sum_i u_{ij}, \tag{39}$$

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

Proof. When $\mathbf{S}^2 = s^2 \mathbf{I}$, Eq. 10 becomes

$$\dot{\boldsymbol{\alpha}} \approx \eta \left(-\log \boldsymbol{\alpha} - \frac{1}{\lambda} \boldsymbol{\alpha} \mathbf{U} \left(s^2 + \lambda \right) \mathbf{I} \mathbf{U}^T \right), \tag{40}$$

$$\dot{\boldsymbol{\alpha}} \approx \eta \left(-\log \boldsymbol{\alpha} - \frac{1}{\lambda} \boldsymbol{\alpha} \left(s^2 + \lambda \right) \right). \tag{41}$$

Fixed points of this system are obtained when $\dot{\boldsymbol{\alpha}} = 0$:

$$0 = \eta \left(-\log \boldsymbol{\alpha} - \frac{s^2 + \lambda}{\lambda} \boldsymbol{\alpha} \right), \tag{42}$$

$$\log \boldsymbol{\alpha} = -\frac{s^2 + \lambda}{\lambda} \boldsymbol{\alpha}. \tag{43}$$

The solution of this equation is

$$\alpha_i^* = \frac{\lambda \mathcal{W}(\lambda^{-1} s^2 + 1)}{s^2 + \lambda} \tag{44}$$

With $\mathbf{z}$ given by Eq. 11, we have

$$\mathbf{z} = \frac{s^2 \mathcal{W}(\lambda^{-1} s^2 + 1)}{s^2 + \lambda} \mathbf{U}^T \mathbf{1}, \tag{45}$$

$$z_i^* = \frac{s^2 \mathcal{W}(\lambda^{-1} s^2 + 1)}{s^2 + \lambda} \sum_i u_{ij}. \tag{46}$$

□

C.8 Eq. 13 Uncoupled Case 2

If the matrix $\mathbf{U}$ is a permutation matrix, the fixed points of Eq. 10 are given by,

$$\alpha_i^* = \frac{\lambda \mathcal{W}(\lambda^{-1} s_i^2 + 1)}{s_i^2 + \lambda}, \qquad z_j^* = \frac{s_i^2 \mathcal{W}(\lambda^{-1} s_i^2 + 1)}{s_i^2 + \lambda}. \tag{47}$$

Proof. When $\mathbf{U}$ is a permutation matrix, it can be made an identity matrix with a meaningless reordering of the class labels . Without loss of generality, we therefore consider $\mathbf{U} = \mathbf{I}$

$$\dot{\boldsymbol{\alpha}} \approx \eta \left(-\log \boldsymbol{\alpha} - \frac{1}{\lambda} \boldsymbol{\alpha} \left(\mathbf{S}^2 + \lambda \right) \mathbf{I} \right). \tag{48}$$

Fixed points of this system are obtained when $\dot{\boldsymbol{\alpha}} = 0$

$$0 = \eta \left(-\log \alpha_i - \frac{1}{\lambda} \alpha_i \left(s_i^2 + \lambda \right) \right), \tag{49}$$

$$\log \alpha_i = -\frac{1}{\lambda} \alpha_i \left(s_i^2 + \lambda \right) \tag{50}$$

The solution of this equation is

$$\alpha_i^* = \frac{\lambda \mathcal{W}(\lambda^{-1} s_i^2 + 1)}{s_i^2 + \lambda}. \tag{51}$$

With $\mathbf{z}$ given by Eq. 11, we have

$$\mathbf{z}^* = \frac{1}{\lambda} \mathbf{S}^2 \mathbf{I} \boldsymbol{\alpha}^{*T}, \tag{52}$$

$$z_j^* = \frac{s_i^2 \mathcal{W}(\lambda^{-1} s_i^2 + 1)}{s_i^2 + \lambda}. \tag{53}$$

□

C.9 Lemma 1 Perturbation Solution

Proof. Starting from the autonomous system Eq. 10 and assumption in 1, we have

$$\boldsymbol{\alpha} = \eta \left(-\log \boldsymbol{\alpha} - \boldsymbol{\alpha} \mathbf{A} \right) \tag{54}$$

where

$$\mathbf{A} = \lambda^{-1} \mathbf{U} (\mathbf{S^2} + \lambda \mathbf{I}) \mathbf{U}^T \tag{55}$$

Since the off-diagonal terms are of order δ, we treat them as a perturbation. The unperturbed system has a solution $\boldsymbol{\alpha}_0$ given by case 2

$$\alpha_{0,i}^* = \frac{\mathcal{W}(A_{ii})}{A_{ii}}. \tag{56}$$

We can linearize the autonomous system Eq. 10 around the unperturbed solution to find,

$$\dot{\boldsymbol{\alpha}} \simeq \eta \left(-\log\left(\boldsymbol{\alpha}_0^*\right) - \boldsymbol{\alpha}_0^* \odot \operatorname{diag}\left(\boldsymbol{A}\right) + \frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\alpha}} \left[-\log\left(\boldsymbol{\alpha}\right) - \boldsymbol{\alpha} \odot \operatorname{diag}\left(\boldsymbol{A}\right) \right] \Big|_{\boldsymbol{\alpha}_0^*} \left(\boldsymbol{\alpha} - \boldsymbol{\alpha}_0^*\right) \right), \tag{57}$$

$$\dot{\boldsymbol{\alpha}} \simeq \eta \left(1 - \log\left(\boldsymbol{\alpha}^*\right) - \left(\operatorname{diag}\left(\boldsymbol{A}\right) + \boldsymbol{\alpha}_0^{*-1} \right) \odot \boldsymbol{\alpha} \right). \tag{58}$$

We then apply the perturbation given by off-diagonal terms of $\mathbf{A}$ to obtain

$$\dot{\boldsymbol{\alpha}} \simeq \eta \left(1 - \log\left(\boldsymbol{\alpha}^*\right) - \boldsymbol{\alpha} \left[\mathbf{A} + \operatorname{diag}\left(\boldsymbol{\alpha}_0^{*-1} \right) \right] \right), \tag{59}$$

where $\operatorname{diag}\left(\boldsymbol{\alpha}_0^{*-1}\right)$ is the diagonal matrix obtained from $\boldsymbol{\alpha}_0^{*-1}$ and where the inverse is applied element by element.

Solving for $\dot{\boldsymbol{\alpha}} = 0$, we obtain the solution

$$\boldsymbol{\alpha}^* = \left(1 - \log\left(\boldsymbol{\alpha}_0^*\right)\right) \left[\mathbf{A} + \operatorname{diag}\left(\boldsymbol{\alpha}_0^{*-1} \right) \right]^{-1}. \tag{60}$$

□

C.10 Thm. 2 Gradient Starvation Regime

Theorem (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,*

$$\frac{\mathrm{d}z_2^*}{\mathrm{d}s_1^2} < 0, \tag{61}$$

which implies GS.

Proof. From lemma 1, and with $\mathbf{U}$ given by Eq. 15, we find that the perturbatory solution for the fixed point is

$$\begin{aligned}
\alpha_1^* = \Bigg[& \lambda \left(W\left(\frac{\lambda + \delta^2\left(s_1^2 - s_2^2\right) + s_2^2}{\lambda} \right) + 1 \right) \Bigg(\delta\sqrt{1-\delta^2}\left(s_2^2 - s_1^2\right) + \lambda e^{W\left(\frac{\lambda + \delta^2\left(s_1^2 - s_2^2\right) + s_2^2}{\lambda} \right)} \\
& \left(W\left(\frac{\lambda + \delta^2\left(-s_1^2\right) + \delta^2 s_2^2 + s_1^2}{\lambda} \right) + 1 \right) \Bigg) \Bigg] \delta^2\left(\delta^2 - 1\right)\left(s_1^2 - s_2^2\right)^2 + \\
& \left(\lambda + \delta^2\left(s_1^2 - s_2^2\right) + \lambda e^{W\left(\frac{\lambda + \delta^2\left(s_1^2 - s_2^2\right) + s_2^2}{\lambda} \right)} + s_2^2 \right) \\
& \left(\lambda + \delta^2 s_2^2 - \left(\delta^2 - 1\right) s_1^2 + \lambda e^{W\left(\frac{\lambda + \delta^2\left(-s_1^2\right) + \delta^2 s_2^2 + s_1^2}{\lambda} \right)} \right) \Bigg]^{-1} \\
\alpha_2^* = \Bigg[& \lambda \left(W\left(\frac{\lambda + \delta^2\left(-s_1^2\right) + \delta^2 s_2^2 + s_1^2}{\lambda} \right) + 1 \right) \Bigg(\delta\sqrt{1-\delta^2}\left(s_2^2 - s_1^2\right) + \lambda e^{W\left(\frac{\lambda + \delta^2\left(-s_1^2\right) + \delta^2 s_2^2 + s_1^2}{\lambda} \right)} \\
& \left(W\left(\frac{\lambda + \delta^2\left(s_1^2 - s_2^2\right) + s_2^2}{\lambda} \right) + 1 \right) \Bigg) \Bigg] \\
& \Bigg[\delta^2\left(\delta^2 - 1\right)\left(s_1^2 - s_2^2\right)^2 + \left(\lambda + \delta^2\left(s_1^2 - s_2^2\right) + \lambda e^{W\left(\frac{\lambda + \delta^2\left(s_1^2 - s_2^2\right) + s_2^2}{\lambda} \right)} + s_2^2 \right) \\
& \left(\lambda + \delta^2 s_2^2 - \left(\delta^2 - 1\right) s_1^2 + \lambda e^{W\left(\frac{\lambda + \delta^2\left(-s_1^2\right) + \delta^2 s_2^2 + s_1^2}{\lambda} \right)} \right) \Bigg]^{-1}
\end{aligned}$$

We have found at Eq. 11 that the corresponding steady-state feature response is given by

$$\mathbf{z}^* = \frac{1}{\lambda}\mathbf{S}^2\mathbf{U}^T\boldsymbol{\alpha}^{*T} \tag{62}$$

In the perturbatory regime δ is taken to be a small parameter. We therefore perform a first-order Taylor series expansion of $\mathbf{z}^*$ around $\delta = 0$ to obtain

$$z_1^* = \frac{\delta s_1^2\left(W\left(\frac{\lambda+s_2^2}{\lambda}\right)+1\right)\left(\lambda+\lambda e^{W\left(\frac{\lambda+s_1^2}{\lambda}\right)}+s_2^2\right)}{\left(\lambda+\lambda e^{W\left(\frac{\lambda+s_1^2}{\lambda}\right)}+s_1^2\right)\left(\lambda+\lambda e^{W\left(\frac{\lambda+s_2^2}{\lambda}\right)}+s_2^2\right)} + \frac{\lambda s_1^2 e^{W\left(\frac{\lambda+s_2^2}{\lambda}\right)}\left(W\left(\frac{\lambda+s_1^2}{\lambda}\right)+1\right)\left(W\left(\frac{\lambda+s_2^2}{\lambda}\right)+1\right)}{\left(\lambda+\lambda e^{W\left(\frac{\lambda+s_1^2}{\lambda}\right)}+s_1^2\right)\left(\lambda+\lambda e^{W\left(\frac{\lambda+s_2^2}{\lambda}\right)}+s_2^2\right)} \tag{63}$$

$$z_2^* = \frac{\lambda s_2^2 e^{W\left(\frac{\lambda+s_1^2}{\lambda}\right)}\left(W\left(\frac{\lambda+s_1^2}{\lambda}\right)+1\right)\left(W\left(\frac{\lambda+s_2^2}{\lambda}\right)+1\right)}{\left(\lambda+\lambda e^{W\left(\frac{\lambda+s_1^2}{\lambda}\right)}+s_1^2\right)\left(\lambda+\lambda e^{W\left(\frac{\lambda+s_2^2}{\lambda}\right)}+s_2^2\right)} - \frac{\delta s_2^2\left(W\left(\frac{\lambda+s_1^2}{\lambda}\right)+1\right)\left(\lambda+\lambda e^{W\left(\frac{\lambda+s_2^2}{\lambda}\right)}+s_1^2\right)}{\left(\lambda+\lambda e^{W\left(\frac{\lambda+s_1^2}{\lambda}\right)}+s_1^2\right)\left(\lambda+\lambda e^{W\left(\frac{\lambda+s_2^2}{\lambda}\right)}+s_2^2\right)} \tag{64}$$

Taking the derivative of z_2^* with respect to s_1, we find

$$\frac{\mathrm{d}z_2^*}{\mathrm{d}s_1^2} = -\frac{\delta\lambda s_2^2\left(e^{W\left(\frac{\lambda+s_1^2}{\lambda}\right)} - e^{W\left(\frac{\lambda+s_2^2}{\lambda}\right)}\right)\left(W\left(\frac{\lambda+s_1^2}{\lambda}\right)+1\right)}{\left(\lambda+\lambda e^{W\left(\frac{\lambda+s_1^2}{\lambda}\right)}+s_1^2\right)^2\left(\lambda+\lambda e^{W\left(\frac{\lambda+s_2^2}{\lambda}\right)}+s_2^2\right)} \tag{65}$$

Knowing that the exponential of the W Lambert function is a strictly increasing function and that $s_1^2 > s_2^2$, we find

$$\frac{\mathrm{d}z_2^*}{\mathrm{d}s_1^2} < 0. \tag{66}$$

□

C.11 Eq. 18 Spectral Decoupling

SD replaces the general L2 weight decay term in Eq. 5 with an L2 penalty exclusively on the network's logits, yielding

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

The loss can be written as,

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}) &= \mathbf{1}\cdot\log\left[1+\exp\left(-\mathbf{Y}\hat{\mathbf{y}}\right)\right] + \frac{\lambda}{2}\|\boldsymbol{\Phi}\boldsymbol{\theta}\|^2,\\ &= \mathbf{1}\cdot\log\left[1+\exp\left(-\mathbf{Y}\hat{\mathbf{y}}\right)\right] + \frac{\lambda}{2}\left\|\boldsymbol{S}\boldsymbol{V}^T\boldsymbol{\theta}\right\|^2,\\ &\geq \mathbf{1}\cdot\left[H(\boldsymbol{\alpha}) - \boldsymbol{\alpha}\boldsymbol{Y}\hat{\mathbf{y}}\right] + \frac{\lambda}{2}\left\|\boldsymbol{S}\boldsymbol{V}^T\boldsymbol{\theta}\right\|^2.\end{aligned}$$

Optimizing $\mathcal{L}(\boldsymbol{\theta})$ wrt to $\boldsymbol{\theta}$ results in the following optimum,

$$\boldsymbol{\theta}^{*T} = \frac{1}{\lambda}\boldsymbol{\alpha}\mathbf{Y}\boldsymbol{\Phi}_0\boldsymbol{V}\boldsymbol{S}^{-2}\boldsymbol{V}^T,$$

which by substitution into the loss function, the dynamics of gradient ascent leads to,

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

where $\log$ and division are taken element-wise on the coordinates of $\boldsymbol{\alpha}$ and hence dynamics of each α_i is independent of other $\alpha_{j\neq i}$.