Robust Learning with Progressive Data Expansion Against Spurious Correlation

Yihe Deng* **Yu Yang*** **Baharan Mirzasoleiman** **Quanquan Gu**
Department of Computer Science
University of California, Los Angeles
Los Angeles, CA 90095
{yihedeng,yuyang,baharan,qgu}@cs.ucla.edu

Abstract

While deep learning models have shown remarkable performance in various tasks, they are susceptible to learning non-generalizable *spurious features* rather than the core features that are genuinely correlated to the true label. In this paper, beyond existing analyses of linear models, we theoretically examine the learning process of a two-layer nonlinear convolutional neural network in the presence of spurious features. Our analysis suggests that imbalanced data groups and easily learnable spurious features can lead to the dominance of spurious features during the learning process. In light of this, we propose a new training algorithm called **PDE** that efficiently enhances the model's robustness for a better worst-group performance. PDE begins with a group-balanced subset of training data and progressively expands it to facilitate the learning of the core features. Experiments on synthetic and real-world benchmark datasets confirm the superior performance of our method on models such as ResNets and Transformers. On average, our method achieves a 2.8% improvement in worst-group accuracy compared with the state-of-the-art method, while enjoying up to $10\times$ faster training efficiency. Codes are available at `https://github.com/uclaml/PDE`.

1 Introduction

Despite the remarkable performance of deep learning models, recent studies (Sagawa et al., 2019, 2020; Izmailov et al., 2022; Haghtalab et al., 2022; Yang et al., 2022, 2023b,c) have identified their vulnerability to spurious correlations in data distributions. A spurious correlation refers to an easily learned feature that, while unrelated to the task at hand, appears with high frequency within a specific class. For instance, waterbirds frequently appear with water backgrounds, and landbirds with land backgrounds. When training with empirical risk minimization (ERM), deep learning models tend to exploit such correlations and fail to learn the more subtle features genuinely correlated with the true labels, resulting in poor generalization performance on minority data (e.g., waterbirds with land backgrounds as shown in Figure 1). This observation raises a crucial question: *Does the model genuinely learn to classify birds, or does it merely learn to distinguish land from water?* The issue is particularly concerning because deep learning models are being deployed in critical applications such as healthcare, finance, and autonomous vehicles, where we require a reliable predictor.

Researchers formalized the problem by considering examples with various combinations of core features (e.g., landbird/waterbird) and spurious features (e.g., land/water backgrounds) as different *groups*. The model is more likely to make mistakes on certain groups if it learns the spurious feature. The objective therefore becomes balancing and improving performance across all groups. Under this formulation, we can divide the task into two sub-problems: (1) accurately identifying the groups, which are not always known in a dataset, and (2) effectively using the group information

*Equal contribution

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

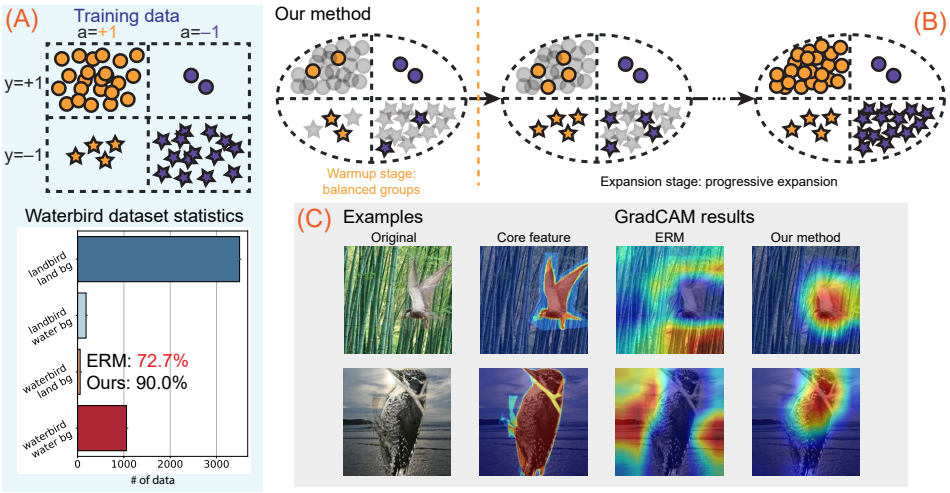


Figure 1: A overview of the problem, our proposed solution, and the resultant outcomes. (A) We demonstrate the data distribution and provide an example of the statistics of Waterbirds. (B) The overall procedure of PDE. (C) we use GradCAM (Selvaraju et al., 2017) to show the attention of the model trained with PDE as compared to ERM. While ERM focuses on the background, PDE successfully trains the model to capture the birds.

to finally improve the model's robustness. While numerous recent works (Nam et al., 2020; Liu et al., 2021; Creager et al., 2021; Ahmed et al., 2021; Taghanaki et al., 2021; Zhang et al., 2022) focus on the first sub-problem, the second sub-problem remains under studied. The pioneering work (Sagawa et al., 2019) still serves as the best guidance for utilizing accurate group information. In this paper, we focus on the second sub-problem and aim to provide a more effective and efficient algorithm to utilize the group information. It is worth noting that the theoretical understanding of spurious correlations lags behind the empirical advancements of mitigating spurious features. Existing theoretical studies (Sagawa et al., 2020; Chen et al., 2020; Yang et al., 2022; Ye et al., 2022) are limited to the setting of simple linear models and data distribution that are less reflective of real application scenarios.

We begin by theoretically examining the learning process of spurious features when training a two-layer nonlinear convolutional neural network (CNN) on a corresponding data model that captures the influence of spurious correlations. We illustrate that the learning of spurious features swiftly overshadows the learning of core features from the onset of training when groups are imbalanced and spurious features are more easily learned than core features. Based upon our theoretical understanding, we propose Progressive Data Expansion (**PDE**), a neat and novel training algorithm that efficiently uses group information to enhance model's robustness against spurious correlations. Existing approaches, such as GroupDRO (Sagawa et al., 2019) and upsampling techniques (Liu et al., 2021), aim to balance the data groups in each batch throughout the training process. In contrast, we employ a small balanced warm-up subset only at the beginning of the training. Following a brief period of balanced training, we progressively expand the warm-up subset by adding small random subsets of the remaining training data until using all of them, as shown in the top right of Figure 1. Here, we utilize the momentum from the warm-up subset to prevent the model from learning spurious features when adding new data. Empirical evaluations on both synthetic and real-world benchmark data validate our theoretical findings and confirm the effectiveness of PDE. Additional ablation studies also demonstrate the significance and impact of each component within our training scheme. In summary, our contributions are highlighted as follows:

- We provide a theoretical understanding of the impact of spurious correlations beyond the linear setting by considering a two-layer nonlinear CNN.
- We introduce PDE, a theory-inspired approach that effectively addresses the challenge posed by spurious correlations.
  - PDE achieves the best performance on benchmark vision and language datasets for models including ResNet and Transformer. On average, it outperforms the state-of-the-art method by 2.8% in terms of worst-group accuracy.
  - PDE enjoys superior training efficiency, being $10\times$ faster than the state-of-the-art methods.

2 Why is Spurious Correlation Harmful to ERM?

In this section, we simplify the intricate real-world problem of spurious correlations into a theoretical framework. We provide analysis on two-layer nonlinear CNNs, extending beyond the linear setting prevalent in existing literature on this subject. Under this framework, we formally present our theory concerning the training process of empirical risk minimization (ERM) in the presence of spurious features. These theoretical insights motivate the design of our algorithm.

2.1 Empirical Risk Minimization

We begin with the formal definition of the ERM-based training objective for a binary classification problem. Consider a training dataset $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, where $\mathbf{x}_i \in \mathbb{R}^d$ is the input and $y \in \{\pm 1\}$ is the output label. We train a model $f(\mathbf{x}; \mathbf{W})$ with weight $\mathbf{W}$ to minimize the empirical loss function:

$$\mathcal{L}(\mathbf{W}) = \frac{1}{N} \sum_{i=1}^N \ell\big(y_i f(\mathbf{x}_i; \mathbf{W})\big), \tag{2.1}$$

where ℓ is the logistic loss defined as $\ell(z) = \log(1 + \exp(-z))$. The empirical risk minimizer refers to $\mathbf{W}^*$ that minimizes the empirical loss: $\mathbf{W}^* := \operatorname{argmin}_{\mathbf{W}} \mathcal{L}(\mathbf{W})$. Typically, gradient-based optimization algorithms are employed for ERM. For example, at each iteration t, gradient descent (GD) has the following update rule:

$$\mathbf{W}^{(t+1)} = \mathbf{W}^{(t)} - \eta \nabla \mathcal{L}(\mathbf{W}^{(t)}). \tag{2.2}$$

Here, $\eta > 0$ is the learning rate. In the next subsection, we will show that even for a relatively simple data model which consists of core features and spurious features, vanilla ERM will fail to learn the core features that are correlated to the true label.

2.2 Data Distribution with Spurious Correlation Fails ERM

Previous work such as (Sagawa et al., 2020) considers a data model where the input consists of core feature, spurious feature and noise patches at fixed positions, i.e., $\mathbf{x} = [\mathbf{x}_{\text{core}}, \mathbf{x}_{\text{spu}}, \mathbf{x}_{\text{noise}}]$. In real-world applications, however, features in an image do not always appear at the same pixels. Hence, we consider a more realistic data model where the patches do not appear at fixed positions.

Definition 2.1 (Data model)**.** A data point $(\mathbf{x}, y, a) \in (\mathbb{R}^d)^P \times \{\pm 1\} \times \{\pm 1\}$ is generated from the distribution $\mathcal{D}$ as follows.

- Randomly generate the true label $y \in \{\pm 1\}$.
- Generate spurious label $a \in \{\pm y\}$, where $a = y$ with probability $\alpha > 0.5$.
- Generate $\mathbf{x}$ as a collection of P patches: $\mathbf{x} = (\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(P)}) \in (\mathbb{R}^d)^P$, where
  - **Core feature.** One and only one patch is given by $\beta_c \cdot y \cdot \mathbf{v}_c$ with $\|\mathbf{v}_c\|_2 = 1$.
  - **Spurious feature.** One and only one patch is given by $\beta_s \cdot a \cdot \mathbf{v}_s$ with $\|\mathbf{v}_s\|_2 = 1$ and $\langle \mathbf{v}_c, \mathbf{v}_s \rangle = 0$.
  - **Random noise.** The rest of the $P - 2$ patches are Gaussian noises $\boldsymbol{\xi}$ that are independently drawn from $N(0, (\sigma_p^2/d) \cdot \mathbf{I}_d)$ with σ_p as an absolute constant.

  And $0 < \beta_c \ll \beta_s \in \mathbb{R}$.

Similar data models have also been considered in recent works on feature learning (Allen-Zhu & Li, 2020; Zou et al., 2021; Chen et al., 2022; Jelassi & Li, 2022), where the input data is partitioned into feature and noise patches. We extend their data models by further positing that certain feature patches might be associated with the spurious label instead of the true label. In the rest of the paper, we assume $P = 3$ for simplicity. With the given data model, we consider the training dataset $S = \{(\mathbf{x}_i, y_i, a_i)\}_{i=1}^N$ and let S be partitioned into large group S_1 and small group S_2 such that S_1 contains all the training data that can be correctly classified by the spurious feature, i.e., $a_i = y_i$, and S_2 contains all the training data that can only be correctly classified by the core feature, i.e., $a_i = -y_i$. We denote $\widehat{\alpha} = \frac{|S_1|}{N}$ and therefore $1 - \widehat{\alpha} = \frac{|S_2|}{N}$.

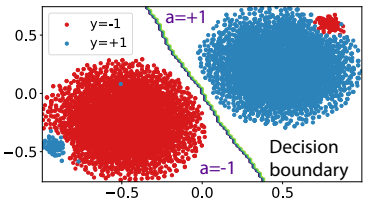


Figure 2: Visualization of the data.

Visualization of our data. In Figure 2, we present the visualization in 2D space of the higher-dimensional data generated from our data model using t-SNE (Van der Maaten & Hinton, 2008), where data within each class naturally segregate into large and small groups. The spurious feature is sufficient for accurate classification of the larger group data, but will lead to misclassification of the small group data.

2.3 Beyond Linear Models

We consider a two-layer nonlinear CNN defined as follows:

$$f(\mathbf{x};\mathbf{W}) = \sum_{j\in[J]}\sum_{p=1}^{P}\sigma\big(\langle \mathbf{w}_j, \mathbf{x}^{(p)}\rangle\big), \tag{2.3}$$

where $\mathbf{w}_j \in \mathbb{R}^d$ is the weight vector of the j-th filter, J is the number of filters (neurons) of the network, and $\sigma(z) = z^3$ is the activation function. $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_J] \in \mathbb{R}^{d\times J}$ denotes the weight matrix of the CNN. Similar two-layer CNN architectures are analyzed in in (Chen et al., 2022; Jelassi & Li, 2022) but for different problems, where the cubic activation serves a simple function that provides non-linearity. Similar to Jelassi & Li (2022); Cao et al. (2022), we assume a mild overparameterization of the CNN with $J = \text{polylog}(d)$. We initialize $\mathbf{W}^{(0)} \sim \mathcal{N}(0, \sigma_0^2)$, where $\sigma_0^2 = \frac{\text{polylog}(d)}{d}$. Due to the CNN structure, our analysis can handle data models where each data can have arbitrary order of patches while linear models fail to do so.

2.4 Understanding the Training Process with Spurious Correlation

In this subsection, we formally introduce our theoretical result on the training process of the two-layer CNN using gradient descent in the presence of spurious features. We first define the performance metrics. A frequenly considered metric is the test accuracy: $\text{Acc}(\mathbf{W}) = \mathbb{P}_{(\mathbf{x},y,a)\sim\mathcal{D}}\big[\text{sgn}(f(\mathbf{x};\mathbf{W})) = y\big]$. With spurious correlations, researchers are more interested in the worst-group accuracy:

$$\text{Acc}_{\text{wg}}(\mathbf{W}) = \min_{y\in\{\pm1\}, a\in\{\pm1\}} \mathbb{P}_{(\mathbf{x},y,a)\sim\mathcal{D}}\big[\text{sgn}(f(\mathbf{x};\mathbf{W})) = y\big],$$

which accesses the worst accuracy of a model among all groups defined by combinations of y and a. We then summarize the learning process of ERM in the following theorem. Our analysis focuses on the learning of spurious and core features, represented by the growth of $\langle \mathbf{w}_i^{(t)}, \mathbf{v}_s\rangle$ and $\langle \mathbf{w}_i^{(t)}, \mathbf{v}_c\rangle$ respectively:

Theorem 2.2. Consider the training dataset $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ that follows the distribution in Definition 2.1. Consider the two-layer nonlinear CNN model as in (2.3) initialized with $\mathbf{W}^{(0)} \sim \mathcal{N}(0, \sigma_0^2)$. After training with GD in (2.2) for $T_0 = \widetilde{\Theta}\big(1/(\eta\beta_s^3\sigma_0)\big)$ iterations, for all $j \in [J]$ and $t \in [0, T_0)$, we have

$$\widetilde{\Theta}(\eta)\beta_s^3(2\widehat{\alpha}-1)\cdot\langle \mathbf{w}_j^{(t)}, \mathbf{v}_s\rangle^2 \le \langle \mathbf{w}_j^{(t+1)}, \mathbf{v}_s\rangle - \langle \mathbf{w}_j^{(t)}, \mathbf{v}_s\rangle \le \widetilde{\Theta}(\eta)\beta_s^3\widehat{\alpha}\cdot\langle \mathbf{w}_j^{(t)}, \mathbf{v}_s\rangle^2, \tag{2.4}$$

$$\widetilde{\Theta}(\eta)\beta_c^3\widehat{\alpha}\cdot\langle \mathbf{w}_j^{(t)}, \mathbf{v}_c\rangle^2 \le \langle \mathbf{w}_j^{(t+1)}, \mathbf{v}_c\rangle - \langle \mathbf{w}_j^{(t)}, \mathbf{v}_c\rangle \le \widetilde{\Theta}(\eta)\beta_c^3\cdot\langle \mathbf{w}_j^{(t)}, \mathbf{v}_c\rangle^2. \tag{2.5}$$

After training for T_0 iterations, with high probability, the learned weight has the following properties: (1) it learns the spurious feature $\mathbf{v}_s$: $\max_{j\in[J]}\langle \mathbf{w}_j^{(T)}, \mathbf{v}_s\rangle \ge \widetilde{\Omega}(1/\beta_s)$; (2) it *almost* does not learn the core feature $\mathbf{v}_c$: $\max_{j\in[J]}\langle \mathbf{w}_j^{(T)}, \mathbf{v}_c\rangle = \widetilde{\mathcal{O}}(\sigma_0)$.

Discussion. The detailed proof is deferred to Appendix E, and we provide intuitive explanations of the theorem as follows. A larger value of $\langle \mathbf{w}_i^{(t)}, \mathbf{v}\rangle$ for $\mathbf{v} \in \{\mathbf{v}_s, \mathbf{v}_c\}$ implies better learning of the feature vector $\mathbf{v}$ by neuron $\mathbf{w}_i$ at iteration t. As illustrated in (2.4) and (2.5), the updates for both spurious and core features are non-zero, as they depend on the squared terms of themselves with non-zero coefficients, while the growth rate of $\langle \mathbf{w}_i^{(t)}, \mathbf{v}_s\rangle$ is significantly faster than that of $\langle \mathbf{w}_i^{(t)}, \mathbf{v}_c\rangle$. Consequently, the neural network rapidly learns the spurious feature but barely learns the core feature, as it remains almost unchanged from initialization.

We derive the neural network's prediction after training for T_0 iterations. For a randomly generated data example $(\mathbf{x}, y, a) \sim \mathcal{D}$, the neural network's prediction is given by $\text{sgn}\big(f(\mathbf{x};\mathbf{W})\big) = \text{sgn}\big(\sum_{j\in[J]}\big(y\beta_c^3\langle \mathbf{w}_j, \mathbf{v}_c\rangle^3 + a\beta_s^3\langle \mathbf{w}_j, \mathbf{v}_s\rangle^3 + \langle \mathbf{w}_j, \xi\rangle^3\big)\big)$. Since the term $\beta_s^3\max_{j\in[J]}\langle \mathbf{w}_j, \mathbf{v}_s\rangle^3$ dominates the summation, the prediction will be $\text{sgn}(f(\mathbf{x};\mathbf{W})) = a$. Consequently, we obtain the test accuracy as $\text{Acc}(\mathbf{W}) = \alpha$, since $a = y$ with probability α, and the model accurately classifies the large group. However, when considering the small group and examining examples where $y \ne a$, the models consistently make errors, resulting in $\text{Acc}_{wg}(\mathbf{W}) = 0$. To circumvent this poor performance on worst-group accuracy, an algorithm that can avoid learning the spurious feature is in demand.

3 Theory-Inspired Two-Stage Training Algorithm

In this section, we introduce Progressive Data Expansion (PDE), a novel two-stage training algorithm inspired by our analysis to enhance robustness against spurious correlations. We begin with illustrating

the implications of our theory, where we provide insights into the data distributions that lead to the rapid learning of spurious features and clarify scenarios under which the model remains unaffected.

3.1 Theoretical Implications

Notably in Theorem 2.2, the growth of the two sequences $\langle \mathbf{w}_i^{(t)}, \mathbf{v}_s \rangle$ in (2.4) and $\langle \mathbf{w}_i^{(t)}, \mathbf{v}_c \rangle$ in (2.5) follows the formula $x_{t+1} = x_t + \eta A x_t^2$, where x_t represents the inner product sequence with regard to iteration t and A is the coefficient containing $\widehat{\alpha}$, β_c or β_s. This formula is closely related to the analysis of tensor power methods (Allen-Zhu & Li, 2020). In simple terms, when two sequences have slightly different growth rates, one of them will experience much faster growth in later times. As we will show below, the key factors that determine the drastic difference of spurious and core features in later times are the group size $\widehat{\alpha}$ and feature strengths β_c, β_s.

- **When the model learns spurious feature ($\beta_c^3 < \beta_s^3(2\widehat{\alpha} - 1)$).** We examine the lower bound for the growth of $\langle \mathbf{w}_i^{(t)}, \mathbf{v}_s \rangle$ in (2.4) and the upper bound for the growth of $\langle \mathbf{w}_i^{(t)}, \mathbf{v}_c \rangle$ in (2.5) in Theorem 2.2. If $\beta_c^3 < \beta_s^3(2\widehat{\alpha} - 1)$, we can employ the tensor power method and deduce that the spurious feature will be learned first and rapidly. The condition on data distribution imposes two necessary conditions: $\widehat{\alpha} > 1/2$ (groups are imbalanced) and $\beta_c < \beta_s$ (the spurious feature is stronger). This observation is consistent with real-world datasets, such as the Waterbirds dataset, where $\widehat{\alpha} = 0.95$ and the background is much easier to learn than the intricate features of the birds.
- **When the model learns core feature ($\beta_c > \beta_s$).** However, if we deviate from the aforementioned conditions and consider $\beta_c > \beta_s$, we can examine the lower bound for the growth of $\langle \mathbf{w}_i^{(t)}, \mathbf{v}_c \rangle$ in (2.5) and the upper bound for the growth of $\langle \mathbf{w}_i^{(t)}, \mathbf{v}_s \rangle$ in (2.4). Once again, we apply the tensor power method and determine that the model will learn the core feature rapidly. In real-world datasets, this scenario corresponds to cases where the core feature is not only significant but also easier to learn than the spurious feature. Even for imbalanced groups with $\widehat{\alpha} > 1/2$, the model accurately learns the core feature. Consequently, enhancing the coefficients of the growth of the core feature allows the model to tolerate imbalanced groups. We present verification through synthetic experiments in the next section.

As we will show in the following subsection, we initially break the conditions of learning the spurious feature by letting $\widehat{\alpha} = 1/2$ in a group-balanced data subset. Subsequently, we utilize the momentum to amplify the core feature's coefficient, allowing for tolerance of $\widehat{\alpha} > 1/2$ when adding new data.

3.2 PDE: A Two-Stage Training Algorithm

We present a new algorithm named Progressive Data Expansion (PDE) in Algorithm 1 and explain the details below, which consist of (1) warm-up and (2) expansion stages.

Algorithm 1 Progressive Data Expansion (PDE)

Require: Number of iterations T_0 for warm-up training; number of times K for dataset expansion; number of iterations J for expansion training; number of data m for each expansion; learning rate η; momentum coefficient γ; initialization scale σ_0; training set $S = \{(\mathbf{x}_i, y_i, a_i)\}_{i=1}^n$; model $f_{\mathbf{W}}$.

Initialize $\mathbf{W}^{(0)}$.
Warm-up stage
Divide the S into groups by values of y and a: $S_{y,a} = \{(\mathbf{x}_i, y_i, a_i)\}_{y_i=y, a_i=a}$.
Generate warm-up set S^0 from S by randomly subsampling from each group of S such that $|S^0_{y,a}| = \min_{y',a'} |S_{y',a'}|$ for $y \in \{\pm 1\}$ and $a \in \{\pm 1\}$.
for $t = 0, 1, \ldots, T_0$ **do**
  Compute loss on S^0: $\mathcal{L}_{S^0}(\mathbf{W}^{(t)}) = \frac{1}{|S^0|} \sum_{i \in S^0} \ell(y_i f(x_i; \mathbf{W}^{(t)}))$.
  Update $\mathbf{W}^{(t+1)}$ by (3.1) and (3.2).
end for
Expansion stage
for $k = 1, \ldots, K$ **do**
  Draw m examples ($S_{[m]}$) from S/S^{k-1} and let $S^k = S^{k-1} \cup S_{[m]}$.
  for $t = 1, \ldots, J$ **do**
    Compute loss on S^k: $\mathcal{L}_{S^k}(\mathbf{W}^{(T)}) = \frac{1}{|S^k|} \sum_{i \in S^k} \ell(y_i f(x_i; \mathbf{W}^{(T)}))$, where $T = T_0 + (k-1) * J + t$.
    Update $\mathbf{W}^{(T+1)}$ by (3.1) and (3.2).
  end for
end for
return $\mathbf{W}^{(t)} = \operatorname{argmax}_{\mathbf{W}^{(t')}} \mathrm{Acc}^{\mathrm{val}}_{\mathrm{wg}}(\mathbf{W}^{(t')})$.

As accelerated gradient methods are most commonly used in applications, we jointly consider the property of momentum and our theoretical insights when designing the algorithm. For gradient descent with momentum (GD+M), at each iteration t and with momentum coefficient $\gamma > 0$, it updates as follows

$$\mathbf{g}^{(t+1)} = \gamma \mathbf{g}^{(t)} + (1-\gamma)\nabla \mathcal{L}(\mathbf{W}^{(t)}), \tag{3.1}$$

$$\mathbf{W}^{(t+1)} = \mathbf{W}^{(t)} - \eta \cdot \mathbf{g}^{(t+1)}, \tag{3.2}$$

Warm-up Stage. In this stage, we create a fully balanced dataset S^0, in which each group is randomly subsampled to match the size of the smallest group, and consider it as a warm-up dataset. We train the model on the warm-up dataset for a fixed number of epochs. During this phase, the model is anticipated to accurately learn the core feature without being influenced by the spurious feature. Note that, under our data model, a completely balanced dataset will have $\widehat{\alpha} = 1/2$. We present the following lemma as a theoretical basis for the warm-up stage.

Lemma 3.1. Given the balanced training dataset $S^0 = \{(\mathbf{x}_i, y_i, a_i)\}_{i=1}^{N_0}$ with $\widehat{\alpha} = 1/2$ as in Definition 2.1 and CNN as in (2.3). The gradient on $\mathbf{v}_s$ will be 0 from the beginning of training.

In particular, with $\widehat{\alpha} = 1/2$ we have $|S_1^0| = |S_2^0|$: an equal amount of data is positively correlated with the spurious feature as the data negatively correlated with the spurious feature. In each update, both groups contribute nearly the same amount of spurious feature gradient with different signs, resulting in cancellation. Ultimately, this prevents the model from learning the spurious feature. Detailed proofs can be found in Appendix F.

Expansion Stage. In this stage, we proceed to train the model by incrementally incorporating new data into the training dataset. The rationale for this stage is grounded in the theoretical result by the previous work (Jelassi & Li, 2022) on GD with momentum, which demonstrates that once gradient descent with momentum initially increases its correlation with a feature $\mathbf{v}$, it retains a substantial historical gradient in the momentum containing $\mathbf{v}$. Put it briefly, the initial learning phase has a considerable influence on subsequent training for widely-used accelerated training algorithms. While ERM learns the spurious feature $\mathbf{v}_s$ and momentum does not help, as we will show in synthetic experiments, PDE avoids learning $\mathbf{v}_s$ and learns $\mathbf{v}_c$ in the warm-up stage. This momentum from warm-up, in turn, amplifies the core feature that is present in the gradients of newly added data, facilitating the continued learning of $\mathbf{v}_c$ in the expansion stage. For a specific illustration, the learning of the core feature by GD+M will be

$$\langle \mathbf{w}_j^{(t+1)}, \mathbf{v}_c \rangle = \langle \mathbf{w}_j^{(t)} - \eta\big(\gamma g^{(t)} + (1-\gamma)\nabla_{\mathbf{w}_j}\mathcal{L}(\mathbf{W}^{(t)})\big), \mathbf{v}_c \rangle,$$

where $g^{(t)}$ is the additional momentum as compared to GD with $\gamma = 0$. While the current gradient along $\mathbf{v}_c$ might be small (i.e., β_c), we can benefit from the historical gradient in $g^{(t)}$ to amplify the growth of $\langle \mathbf{w}_j^{(t+1)}, \mathbf{v}_c \rangle$ and make it larger than that of the spurious feature (i.e., β_s). This learning process will then correspond to the case when the model learns the core feature discussed in Subsection 3.1. Practically, we consider randomly selecting m new examples for expansion every J epochs by attempting to draw a similar number of examples from each group. During the last few epochs of the expansion stage, we expect the newly incorporated data exclusively from the larger group, as the smaller groups have been entirely integrated into the warm-up dataset.

It is worth noting that while many works address the issue of identifying groups from datasets containing spurious correlations, we assume the group information is known and our algorithm focuses on the crucial subsequent question of optimizing group information utilization. Aiming to prevent the learning of spurious features, PDE distinguishes itself by employing a rapid and lightweight warm-up stage and ensuring continuous improvement during the expansion stage with the momentum acquired from the warm-up dataset. Our training framework is both concise and effective, resulting in computational efficiency and ease of implementation.

4 Experiments

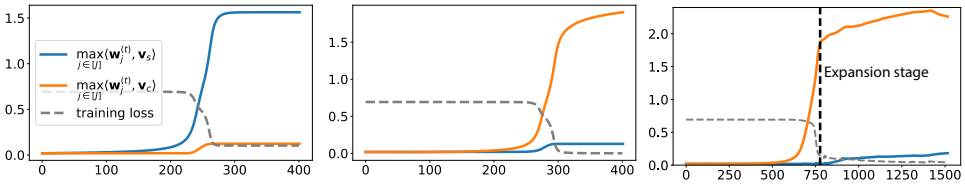
In this section, we present the experiment results from both synthetic and real datasets. Notably, we report the worst-group accuracy, which assesses the minimum accuracy across all groups and is commonly used to evaluate the model's robustness against spurious correlations.

4.1 Synthetic Data

In this section, we present synthetic experiment results in verification to our theoretical findings. In Appendix A, we illustrate the detailed data distribution, hyper-parameters of the experiments and more extensive experiment results. The data used in this seciton is generated following Definition 2.1. We consider the worst-group and overall test accuracy. As illustrated in Table 1, ERM, whether trained with GD or GD+M, is unable to accurately predict the small group in our specified data distribution where $\widehat{\alpha} = 0.98$ and $\beta_c < \beta_s$. In contrast, our method significantly improves worst-group accuracy while maintaining overall test accuracy comparable to ERM. Furthermore, as depicted in Figure 3(a), ERM rapidly learns the spurious feature as it minimizes the training loss, while barely learning the core feature. Meanwhile, in Figure 3(b) we show the learning of ERM when the data distribution breaks the conditions of our theory and has $\beta_c > \beta_s$ instead. Even with the same $\widehat{\alpha}$ as in Figure 3(a), ERM correctly learns the core feature despite the imbalanced group size. These two figures support the theoretical results we discussed to motivate our method. Consequently, on the same training dataset as in Figure 3(a), Figure 3(c) shows that our approach allows the model to initially learn the core feature using the warm-up dataset and continue learning when incorporating new data.

Table 1: Synthetic data experiments. We report the worst-group accuracy and the gap (i.e., overall - worst). We further consider several variations of PDE to demonstrate the importance of each component of our method. **Reset**: we reset the momentum to zero after the warm-up stage. **Warmup+All**: we let PDE to incorporate all of the new training data at once after the warm-up stage.

	Worst-group (%)	Gap (%)
ERM (GD)	0.00	97.71
ERM (GD+M)	0.00	97.71
Warmup+All (Reset)	67.69	31.18
Warmup+All	74.24	24.76
PDE (Reset)	92.51	2.29
PDE	**93.01**	**0.03**



(a) ERM (case 1) (b) ERM (case 2) (c) PDE

Figure 3: **Training process of ERM vs. PDE.** We consider the same dataset generated from the distribution as in Definition 2.1 for ERM (case 1) and PDE. On the same training data, ERM learns the spurious feature while PDE successfully learns the core feature. We further consider ERM (case 2) when training on the data distribution where $\beta_c > \beta_s$ and $\widehat{\alpha} = 0.98$. We show the growth of the max inner product between the model's neuron and core/spurious signal vector and the decrease of training loss with regard to the number of iterations t.

4.2 Real Data

We conduct experiments on real benchmark datasets to: (1) compare our approach with state-of-the-art methods, highlighting its superior performance and efficiency, and (2) offer insights into the design of our method through ablation studies.

Datasets. We evaluate on three wildly used datasets across vision and language tasks for spurious correlation: (1) **Waterbirds** (Sagawa et al., 2019) contains bird images labeled as waterbird or landbird, placed against a water or land background, where the smallest subgroup is waterbirds on land background. (2) **CelebA** (Liu et al., 2015) is used to study gender as the spurious feature for hair color classification, and the smallest group in this task is blond-haired males. (3) **CivilComments-WILDS** (Koh et al., 2021b) classifies toxic and non-toxic online comments while dealing with demographic information. It creates 16 overlapping groups for each of the 8 demographic identities.

Baselines. We compare our proposed algorithm against several state-of-the-art methods. Apart from standard ERM, we include GroupDRO (Sagawa et al., 2019) and DFR (Kirichenko et al., 2023) that assume access to the group labels. We only include DFR^{Tr} for fair comparison as all methods only use the training dataset to train and finetune the model, while DFR^{Val} also finetunes on validation data. We also design a baseline called subsample that simply trains the model on the warm-up dataset only. Additionally, we evaluate three recent methods that address spurious correlations without the need for group labels: LfF (Nam et al., 2020), EIIL (Creager et al., 2021), and JTT (Liu et al., 2021). We

Table 2: The worst-group and average accuracy (%) of PDE compared with state-of-the-art methods. The **bold** numbers indicate the best results among the methods that *require group information*, while the underscored numbers represent methods that *only train once*. All methods use validation data for early stopping and model selection, while $\surd\surd$ indicates that the method also re-trains the last layer using the validation data

Method	**Group info**	Train once	Val info	Waterbirds		CelebA		CivilComments	
				Worst	Average	Worst	Average	Worst	Average
ERM	×	$\surd$	$\surd$	$70.0_{\pm 2.3}$	$97.1_{\pm 0.1}$	$45.0_{\pm 1.5}$	$94.8_{\pm 0.2}$	$58.2_{\pm 2.8}$	$92.2_{\pm 0.1}$
LfF	×	×	$\surd$	$78.0_{N/A}$	$91.2_{N/A}$	$77.2_{N/A}$	$85.1_{N/A}$	$58.8_{N/A}$	$92.5_{N/A}$
EIIL	×	×	$\surd$	$77.2_{\pm 1.0}$	$96.5_{\pm 0.2}$	$81.7_{\pm 0.8}$	$85.7_{\pm 0.1}$	$67.0_{\pm 2.4}$	$90.5_{\pm 0.2}$
JTT	×	×	$\surd$	$86.7_{N/A}$	$93.3_{N/A}$	$81.1_{N/A}$	$88.0_{N/A}$	$69.3_{N/A}$	$91.1_{N/A}$
Subsample	$\surd$	$\surd$	$\surd$	$86.9_{\pm 2.3}$	$89.2_{\pm 1.2}$	$86.1_{\pm 1.9}$	$91.3_{\pm 0.2}$	$64.7_{\pm 7.8}$	$83.7_{\pm 3.4}$
DFR^{Tr}	$\surd$	×	$\surd$	$90.2_{\pm 0.8}$	$97.0_{\pm 0.3}$	$80.7_{\pm 2.4}$	$90.6_{\pm 0.7}$	$58.0_{\pm 1.3}$	$92.0_{\pm 0.1}$
DFR^{Val}	$\surd$	×	$\surd\surd$	$\mathbf{92.9}_{\pm 0.2}$	$94.2_{\pm 0.4}$	$88.3_{\pm 1.1}$	$91.3_{\pm 0.3}$	$70.1_{\pm 0.8}$	$87.2_{\pm 0.3}$
GroupDRO	$\surd$	$\surd$	$\surd$	$86.7_{\pm 0.6}$	$93.2_{\pm 0.5}$	$86.3_{\pm 1.1}$	$92.9_{\pm 0.3}$	$69.4_{\pm 0.9}$	$89.6_{\pm 0.5}$
PDE	$\surd$	$\surd$	$\surd$	$\underline{90.3}_{\pm 0.3}$	$92.4_{\pm 0.8}$	$\underline{\mathbf{91.0}}_{\pm 0.4}$	$92.0_{\pm 0.6}$	$\underline{\mathbf{71.5}}_{\pm 0.5}$	$86.3_{\pm 1.7}$

Table 3: Training efficiency of PDE and GroupDRO on Waterbirds. We compare with GroupDRO at their learning rate and weight decay, as well as at ours. We report the worst-group accuracy, average accuracy and the number of epochs till early stopping as the model reached the best performance on validation data. Note: for a fair comparison, we consider one training epoch as training over the N data as the size of the training dataset.

Method	Learning rate	Weight decay	Worst	Average	Early-stopping epoch*
GroupDRO	1e-5	1e-0	$86.7_{\pm 0.6}$	$93.2_{\pm 0.5}$	$92_{\pm 4}$
GroupDRO	1e-2	1e-2	$77.3_{\pm 2.0}$	$97.1_{\pm 0.5}$	$15_{\pm 15}$
PDE	1e-2	1e-2	$\mathbf{90.3}_{\pm 0.3}$	$92.4_{\pm 0.8}$	$8.9_{\pm 1.8}$

report results for ERM, Subsample, GroupDRO and PDE based on our own runs using the WILDS library Koh et al. (2021a); for others, we directly reuse their reported numbers.

We present the experiment details including dataset statistics and hyperparameters as well as comprehensive additional experiments in Appendix B.

4.2.1 Consistent Superior Worst-group Performance

We assess PDE on the mentioned datasets with the state-of-the-art methods. Importantly, we emphasize the comparison with GroupDRO, as it represents the best-performing method that utilizes group information. As shown in Table 2, PDE considerably enhances the worst-performing group's performance across all datasets, while maintaining the average accuracy comparable to GroupDRO. Remarkably, although GroupDRO occasionally fails to surpass other methods, PDE's performance consistently exceeds them in worst-group accuracy.

4.2.2 Efficient Training

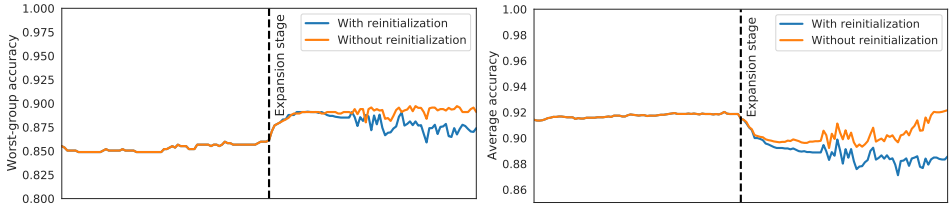
In this subsection, we show that our method is more efficient as it does not train a model twice (as in JTT) and more importantly avoids the necessity for a small learning rate (as in GroupDRO). Specifically, methods employing group-balanced batches like GroupDRO require a very small learning rate coupled with a large weight decay in practice. We provide an intuitive explanation as follows. When sampling to achieve balanced groups in each batch, smaller groups appear more frequently than larger ones. If training progresses rapidly, the loss on smaller groups will be minimized quickly, while the majority of the large group data remains unseen and contributes to most of the gradients in later batches. Therefore, these methods necessitate slow training to ensure the model encounters diverse data from larger groups before completely learning the smaller groups. We validate this observation in Table 3, where GroupDRO trained faster than the default results in significantly poorer performance similar to ERM. Conversely, PDE can be trained to converge rapidly on the warm-up set and reaches better worst-group accuracy $10\times$ faster than GroupDRO at default. Note that methods which only finetune the last layer (Kirichenko et al., 2023; Wei et al., 2023) are also efficient. However, they still require training a model first using ERM on the entire training data till convergence. In contrast, PDE does not require to further finetune the model.

4.2.3 Understanding the Two Stages

We examine each component and demonstrate their effect in PDE. In Table 4, we present the worst-group and average accuracy of the model trained following the warm-up and expansion stages. Indeed, the majority of learning occurs in the warm-up stage, during which a satisfactory worst-group accuracy is established. In the expansion stage, the model persists in learning new data along the established trajectory, leading to continued performance improvement. In Figure 5, we corroborate and emphasize that the model has acquired the appropriate features and maintains learning based on its historical gradient stored in the momentum by presenting the following evidence. As shown, if the optimizer is reset after the warm-up stage and loses all its historical gradients, it rapidly acquires spurious features, resulting in a swift decline in performance accuracy as shown in the blue line.

Table 4: Performance of PDE after each stage. We report the worst-group and average accuracy.

	Warm-up		Addition	
Dataset	Worst	Avg	Worst	Avg
Waterbirds	86.0	91.9	**90.3**	92.4
CelebA	87.8	92.1	**91.0**	92.0
CivilComm	67.7	78.8	**71.5**	86.3

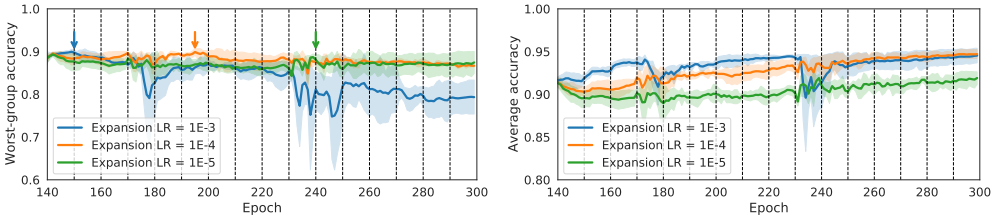


(a) Worst-group accuracy. (b) Average accuracy.

Figure 4: The effect of reset the momentum to zero after the warm-up stage for PDE on Waterbirds.

4.2.4 Ablation Study on the Hyper-parameters of PDE

PDE is robust within a reasonable range of hyperparemeter choices, although some configurations outperform others. As shown in Table 5, it is necessary to limit the number of data points introduced during each expansion to prevent performance degradation. Similarly, in Appendix A, we emphasize the importance of gradual data expansion. In Table 6, we show that post-warmup learning rate decay is essential, though PDE exhibits a tolerance to the degree of this decay. Lastly, as illustrated in Figure 5, adopting a smaller learning rate often necessitates increased data expansions. Nonetheless, a reduced learning rate does not necessarily lead to improved performance.



(a) Worst-group accuracy. (b) Average accuracy.

Figure 5: The variations in both worst-group and average accuracy on the test set of Waterbirds during expansion stage under different expansion learning rates. Each vertical dashed line denotes an expansion and the arrow denotes the early stopping.

Table 5: Ablation study on Waterbirds. Exp. size: number of data points added in each expansion.

Exp. size	Exp. lr	Worst	Average
5	1e-4	$89.9_{\pm 0.5}$	$92.1_{\pm 0.3}$
10	1e-4	$\mathbf{90.3}_{\pm 0.3}$	$92.4_{\pm 0.8}$
50	1e-4	$88.1_{\pm 0.8}$	$93.4_{\pm 0.4}$

Table 6: Ablation study on Waterbirds. Exp. lr: the learning rate in expansion stage.

Exp. size	Exp. lr	Worst	Average
10	1e-2	$85.4_{\pm 3.1}$	$92.1_{\pm 2.0}$
10	1e-3	$89.4_{\pm 0.7}$	$92.6_{\pm 0.3}$
10	1e-5	$89.5_{\pm 0.2}$	$92.1_{\pm 0.1}$

5 Related Work

Existing approaches for improving robustness against spurious correlations can be categorized into two lines of research based on the tackled subproblems. A line of research focuses on the same subproblem we tackle: effectively using the group information to improve robustness. With group information, one can use the distributionally robust optimization (DRO) framework and dynamically increase the weight of the worst-group loss in minimization (Hu et al., 2018; Oren et al., 2019; Sagawa et al., 2019; Zhang et al., 2021). Within this line of work, GroupDRO (Sagawa et al., 2019) achieves state-of-the-art performances across multiple benchmarks. Other approaches use importance weighting to reweight the groups (Shimodaira, 2000; Byrd & Lipton, 2019; Xu et al., 2021) and class balancing to downsample the majority or upsample the minority (He & Garcia, 2009; Cui et al., 2019; Sagawa et al., 2020). Alternatively, Goel et al. (2021) leverage group information to augment the minority groups with synthetic examples generated using GAN. Another strategy (Cao et al., 2019, 2020) involves imposing Lipschitz regularization around minority data points. Most recently, methods that train a model using ERM first and then only finetune the last layer on balanced data from training or validation (Kirichenko et al., 2023) or learn post-doc scaling adjustments (Wei et al., 2023) are shown to be effective.

The other line of research focuses on the setting where group information is not available during training and tackles the first subproblem we identified as accurately finding the groups. Recent notable works (Nam et al., 2020; Liu et al., 2021; Creager et al., 2021; Zhang et al., 2022; Yang et al., 2023a) mostly involve training two models, one of which is used to find group information. To finally use the found groups, many approaches (Namkoong & Duchi, 2017; Duchi et al., 2019; Oren et al., 2019; Sohoni et al., 2020) still follow the DRO framework.

The first theoretical analysis of spurious correlation is provided by Sagawa et al. (2020). For self-supervised learning, Chen et al. (2020) shows that fine-tuning with pre-trained models can reduce the harmful effects of spurious features. Ye et al. (2022) provides guarantees in the presence of label noise that core features are learned well only when less noisy than spurious features. These theoretical work only provides analyses of linear models. Meanwhile, a parallel line of work has established theoretical analysis of nonlinear CNNs in the more realistic setting Allen-Zhu & Li (2020); Zou et al. (2021); Wen & Li (2021); Chen et al. (2022); Jelassi & Li (2022). Our work builds on this line of research and generalizes it to the study of spurious features. Lastly, we notice that a concurrent work (Chen et al., 2023) also uses tensor power method (Allen-Zhu & Li, 2020) to analyze the learning of spurious features v.s. invariant features, but in the setting of out-of-distribution generalization.

6 Conclusion

In conclusion, this paper addressed the challenge of spurious correlations in training deep learning models and focused on the most effective use of group information to improve robustness. We provided a theoretical analysis based on a simplified data model and a two-layer nonlinear CNN. Building upon this understanding, we proposed PDE, a novel training algorithm that effectively and efficiently enhances model robustness against spurious correlations. This work contributes to both the theoretical understanding and practical application of mitigating spurious correlations, paving the way for more reliable and robust deep learning models.

Limitations and future work. Although beyond the linear setting, our analysis still focuses on a relatively simplified binary classification data model. To better represent real-world application scenarios, future work could involve extending to multi-class classification problems and examining the training of transformer architectures. Practically, our proposed method requires the tuning of additional hyperparameters, including the number of warm-up epochs, the number of times for dataset expansion and the number of data to be added in each expansion.

Acknowledgements

We sincerely thank Dongruo Zhou for the constructive suggestions on the structure and writings of the paper. We also thank the anonymous reviewers for their helpful comments. YD and QG are supported in part by the National Science Foundation CAREER Award 1906169 and IIS-2008981, and the Sloan Research Fellowship. BM is supported by the National Science Foundation CAREER Award 2146492. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

References

Ahmed, F., Bengio, Y., Van Seijen, H., and Courville, A. Systematic generalisation with group invariant predictions. In *International Conference on Learning Representations*, 2021.

Allen-Zhu, Z. and Li, Y. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. *arXiv preprint arXiv:2012.09816*, 2020.

Byrd, J. and Lipton, Z. What is the effect of importance weighting in deep learning? In *International conference on machine learning*, pp. 872–881. PMLR, 2019.

Cao, K., Wei, C., Gaidon, A., Arechiga, N., and Ma, T. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32, 2019.

Cao, K., Chen, Y., Lu, J., Arechiga, N., Gaidon, A., and Ma, T. Heteroskedastic and imbalanced deep learning with adaptive regularization. *arXiv preprint arXiv:2006.15766*, 2020.

Cao, Y., Chen, Z., Belkin, M., and Gu, Q. Benign overfitting in two-layer convolutional neural networks. *Advances in neural information processing systems*, 35:25237–25250, 2022.

Chen, Y., Wei, C., Kumar, A., and Ma, T. Self-training avoids using spurious features under domain shift. *Advances in Neural Information Processing Systems*, 33:21061–21071, 2020.

Chen, Y., Huang, W., Zhou, K., Bian, Y., Han, B., and Cheng, J. Towards understanding feature learning in out-of-distribution generalization. *arXiv preprint arXiv:2304.11327*, 2023.

Chen, Z., Deng, Y., Wu, Y., Gu, Q., and Li, Y. Towards understanding the mixture-of-experts layer in deep learning. *Advances in neural information processing systems*, 2022.

Creager, E., Jacobsen, J.-H., and Zemel, R. Environment inference for invariant learning. In *International Conference on Machine Learning*, pp. 2189–2200. PMLR, 2021.

Cui, Y., Jia, M., Lin, T.-Y., Song, Y., and Belongie, S. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9268–9277, 2019.

Duchi, J. C., Hashimoto, T., and Namkoong, H. Distributionally robust losses against mixture covariate shifts. *Under review*, 2:1, 2019.

Goel, K., Gu, A., Li, Y., and Re, C. Model patching: Closing the subgroup performance gap with data augmentation. In *International Conference on Learning Representations*, 2021.

Haghtalab, N., Jordan, M., and Zhao, E. On-demand sampling: Learning optimally from multiple distributions. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022.

He, H. and Garcia, E. A. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009.

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

Hu, W., Niu, G., Sato, I., and Sugiyama, M. Does distributionally robust supervised learning give robust classifiers? In *International Conference on Machine Learning*, pp. 2029–2037. PMLR, 2018.

Izmailov, P., Kirichenko, P., Gruver, N., and Wilson, A. G. On feature learning in the presence of spurious correlations. *Advances in Neural Information Processing Systems*, 35:38516–38532, 2022.

Jelassi, S. and Li, Y. Towards understanding how momentum improves generalization in deep learning. In *International Conference on Machine Learning*, pp. 9965–10040. PMLR, 2022.

Kirichenko, P., Izmailov, P., and Wilson, A. G. Last layer re-training is sufficient for robustness to spurious correlations. In *International Conference on Learning Representations*, 2023.

Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Gao, I., Lee, T., David, E., Stavness, I., Guo, W., Earnshaw, B. A., Haque, I. S., Beery, S., Leskovec, J., Kundaje, A., Pierson, E., Levine, S., Finn, C., and Liang, P. WILDS: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning (ICML)*, 2021a.

Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Gao, I., et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pp. 5637–5664. PMLR, 2021b.

Liu, E. Z., Haghgoo, B., Chen, A. S., Raghunathan, A., Koh, P. W., Sagawa, S., Liang, P., and Finn, C. Just train twice: Improving group robustness without training group information. In *International Conference on Machine Learning*, pp. 6781–6792. PMLR, 2021.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pp. 3730–3738, 2015.

Nam, J., Cha, H., Ahn, S., Lee, J., and Shin, J. Learning from failure: De-biasing classifier from biased classifier. *Advances in Neural Information Processing Systems*, 33:20673–20684, 2020.

Namkoong, H. and Duchi, J. C. Variance-based regularization with convex objectives. *Advances in neural information processing systems*, 30, 2017.

Oren, Y., Sagawa, S., Hashimoto, T. B., and Liang, P. Distributionally robust language modeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4227–4237, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1432.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. 2019.

Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2019.

Sagawa, S., Raghunathan, A., Koh, P. W., and Liang, P. An investigation of why overparameterization exacerbates spurious correlations. In *International Conference on Machine Learning*, pp. 8346–8356. PMLR, 2020.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

Shimodaira, H. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.

Sohoni, N., Dunnmon, J., Angus, G., Gu, A., and Ré, C. No subclass left behind: Fine-grained robustness in coarse-grained classification problems. *Advances in Neural Information Processing Systems*, 33:19339–19352, 2020.

Taghanaki, S. A., Choi, K., Khasahmadi, A. H., and Goyal, A. Robust representation learning via perceptual similarity metrics. In *International Conference on Machine Learning*, pp. 10043–10053. PMLR, 2021.

Van der Maaten, L. and Hinton, G. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. The caltech-ucsd birds-200-2011 dataset. 2011.

Wei, J., Narasimhan, H., Amid, E., Chu, W.-S., Liu, Y., and Kumar, A. Distributionally robust post-hoc classifiers under prior shifts. In *International Conference on Learning Representations (ICLR)*, 2023.

Wen, Z. and Li, Y. Toward understanding the feature learning process of self-supervised contrastive learning. In *International Conference on Machine Learning*, pp. 11112–11122. PMLR, 2021.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/2020.emnlp-demos.6`.

Xu, D., Ye, Y., and Ruan, C. Understanding the role of importance weighting for deep learning. In *International Conference on Learning Representations*, 2021.

Yang, Y., Gan, E., Dziugaite, G. K., and Mirzasoleiman, B. Identifying spurious biases early in training through the lens of simplicity bias. *arXiv preprint arXiv:2305.18761*, 2023a.

Yang, Y., Nushi, B., Palangi, H., and Mirzasoleiman, B. Mitigating spurious correlations in multi-modal models during fine-tuning. In *International Conference on Machine Learning*, 2023b.

Yang, Y., Zhang, H., Katabi, D., and Ghassemi, M. Change is hard: A closer look at subpopulation shift. *arXiv preprint arXiv:2302.12254*, 2023c.

Yang, Y.-Y., Chou, C.-N., and Chaudhuri, K. Understanding rare spurious correlations in neural networks. *arXiv preprint arXiv:2202.05189*, 2022.

Ye, H., Zou, J., and Zhang, L. Freeze then train: Towards provable representation learning under spurious correlations and feature noise. *arXiv preprint arXiv:2210.11075*, 2022.

Zhang, J., Menon, A. K., Veit, A., Bhojanapalli, S., Kumar, S., and Sra, S. Coping with label shift via distributionally robust optimisation. In *International Conference on Learning Representations*, 2021.

Zhang, M., Sohoni, N. S., Zhang, H. R., Finn, C., and Ré, C. Correct-n-contrast: A contrastive approach for improving robustness to spurious correlations. *arXiv preprint arXiv:2203.01517*, 2022.

Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., and Torralba, A. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

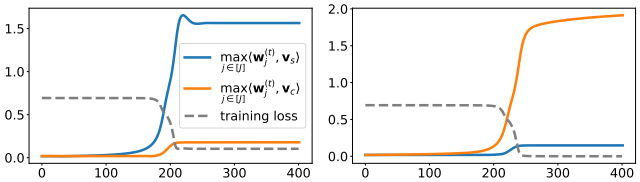
Zou, D., Cao, Y., Li, Y., and Gu, Q. Understanding the generalization of adam in learning neural networks with proper regularization. *arXiv preprint arXiv:2108.11371*, 2021.

A Synthetic Experiments

Datasets. We generate $10,000$ training examples and $10,000$ test examples from the data distribution defined in Definition 2.1 with dimension $d = 50$ and number of patches $P = 3$. Specifically, we let $\alpha = 0.98$, $\beta_c = 0.2$, $\beta_s = 1$ and $\sigma_p = 0.78$ for Table 1 as well as Figure 3(a) and Figure 3(c). For Figure 3(b), we consider a data distribution where $\alpha = 0.98$, $\beta_c = 1$, $\beta_s = 0.2$ and $\sigma_p = 0.78$. Furthermore, we randomly shuffle the order of the patches of $\mathbf{x}$ after we generate data $(\mathbf{x}, y, a)$.

Training. We consider the performances of a nonlinear CNN trained with ERM and PDE. The nonlinear CNN architecture follows (2.3) with the cubic activation function, where we let the number of neurons/filters $J = 40$. We use gradient descent with momentum (GD+M) as the optimizer of our method, setting the momentum to 0.9 and the learning rate to 0.03. The number of warm-up iterations is set to 800. We consider ERM trained with GD with a learning rate 0.1 and without momentum to align with our theoretical finding in both Table 1 and Figure 3. In Table 1, we also show the experiment results for ERM trained with GD+M as same as PDE. All models are trained until convergence.

Additional experiments. In Figure 6, we demonstrate the growth of $\max_{j\in[J]}\langle \mathbf{w}_j^{(t)}, \mathbf{v}_s\rangle$ and $\max_{j\in[J]}\langle \mathbf{w}_j^{(t)}, \mathbf{v}_c\rangle$ for ERM trained with GD+M under the same data generated in Figure 3. Similarly, we observe that ERM learns the spurious feature quickly as the training loss is minimized under our data distribution. Meanwhile, if the data is generated as in case 2 where $\beta_c > \beta_s$, ERM learns the core feature correctly.



(a) ERM (case 1) (b) ERM (case 2)

Figure 6: **Training process of ERM trained with GD+M.** We consider the same dataset generated in Figure 3 and observe almost the same training process as ERM with GD, except GD+M learns the features faster.

Furthermore, we consider the following variation of our methods on the same dataset in Table 1 to demonstrate the importance of gradual expansion. In Figure 7, we let PDE to incorporate all of the new training data at once after warm-up stage. As demonstrated, adding all data at once makes it harder for the model to continue learning core feature, resulting in a worst-group accuracy of 74.24% as compared to 94.32% for progressive expansion.

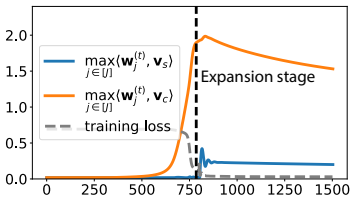


Figure 7: **Variation of PDE.** We consider the same dataset generated in Figure 3 and add all data at once after the warm-up stage.

B Benchmark Datasets

Waterbirds. The Waterbirds dataset (Sagawa et al., 2019) was constructed to study object recognition models relying on image backgrounds instead of the object itself. To this end, bird images from the Caltech-UCSD Birds-200-2011 (CUB) dataset (Wah et al., 2011) were combined with backgrounds from the Places dataset (Zhou et al., 2017). The dataset contains $4,795$ bird images labeled as a waterbird or landbird and placed against a water or land background. Waterbirds are predominantly located against a water background, while landbirds are situated against a land background. Notably, the smallest subgroup in the dataset is waterbirds on land, consisting of only 56 examples.

Table 7: Number of data in our wamr-up dataset for PDE's results in Table 2. We also report the number of data in total for the three datasets.

Dataset	Warm-up	All
Waterbirds	224	4,795
CelebA	5,548	162,770
CivilComments-WILDS	13,705	269,038

CelebA. The CelebA dataset (Liu et al., 2015) is a popular face attribute dataset used to examine the spurious associations between non-demographic and demographic attributes. Specifically, one of the 40 binary attributes, "blond hair", is used as the target attribute, and "male" is the spurious attribute. The dataset contains $162,770$ training examples, with the smallest group being blond-haired males, with only 1387 examples.

CivilComments-WILDS. The CivilComments-WILDS dataset (Koh et al., 2021b) is designed to explore the challenge of classifying online comments as either toxic or non-toxic while dealing with the spurious correlation between the label and demographic information such as gender, race, religion, and sexual orientation. The dataset's evaluation metric, as defined by Koh et al. (2021b), creates 16 overlapping groups for each of the eight demographic identities, resulting in a total of 512 distinct groups. For each group, the metric calculates the worst-case performance of a classifier, which allows for a robust evaluation of the model's ability to generalize across diverse populations.

C Real Data Experiments

Setup. Our experiment settings strictly follow the same setting used for datasets introduced in Appendix B in previous works (Sagawa et al., 2019; Liu et al., 2021; Nam et al., 2020; Creager et al., 2021; Kirichenko et al., 2023). Specifically, we built our training pipeline with the WILDS package (Koh et al., 2021a) which uses pretrained ResNet-50 model (He et al., 2016) in Pytorch (Paszke et al., 2019) library for the image datasets (i.e., Waterbirds and CelebA) and Transformer (Vaswani et al., 2017) in Transformers library (Wolf et al., 2020) for CivilComments-WILDS. All experiments were conducted on a single NVIDIA RTX A6000 GPU with 48GB memory.

Training. In Table 7, we summarize the number of data used in warm-up stage for PDE in Table 2 with the total number of data in the entire datasets. In Table 8, we report the hyperparameters used for PDE with the notations in Algorithm 1. Specifically, T_0 refers to the number of epochs for warm-up stage and J refers to the number of epochs for training after each data expansion. Lastly, m is the number of added data for each data expansion. Our batch size is consistent with GroupDRO.

Table 8: Hyperparameters used for PDE's results in Table 2. Note that T_0 and J are in epochs of PDE's training set, which have fewer iterations than epochs of the full training set.

Dataset	Learning rate	Weight decay	Batch size	T_0	J	m
Waterbirds	1e-2	1e-2	64	140	10	10
CelebA	1e-2	1e-4	128	16	10	50
CivilComments-WILDS	1e-5	1e-2	16	15	2	300

Groups for CivilComments-WILDS. We note that the demographic tags in CivilComments-WILDS can coexist in the input text. For example, a text can contain both tags of female and male. Therefore, combining the 8 demographic tags with the binary classification label (toxic vs. non-toxic) results in 16 overlapping groups, where each group counts as data from a class with/without a specific tag. For computational efficiency, previous methods divide the data into four non-overlapping groups either by the *specific* one demographic tag a_i (groups are $\{y = \pm 1, a_i = \pm 1\}$) (Koh et al., 2021b) or by containing *any* one of the tags: $a = 1$ if any $a_i = 1$ and $a = -1$ otherwise (groups are $\{y = \pm 1, a = \pm 1\}$) (Liu et al., 2021; Creager et al., 2021). However, the data can actually be partitioned into 512 distinct groups, with each group corresponding to different combinations of tags: $\{y = \pm 1, a_1 = \pm 1, a_2 = \pm 1, \ldots, a_n = \pm 1\}$. As GroupDRO requires computation per group at each training batch, considering a large number of groups makes it harder for GroupDRO to train efficiently. Meanwhile, having more groups does not impose an additional computational cost on PDE, so we can consider all these data groups when constructing our warm-up set. As many groups

are empty or contain very little data, we set a threshold to select at most 150 data points from each group to ensure a balanced yet sufficient warm-up set.

Efficiency. In Table 9, we further report the training efficiency of PDE compared with GroupDRO on CelebA and CivilComments-WILDS. As similar to what we observe on Waterbirds dataset, PDEachieves best performance at a larger learning rate and smaller weight decay on CelebA with a significant speedup as compared to GroupDRO. On CivilComments-WILDS, we can also observe an improved efficiency.

Table 9: Training efficiency of PDE and GroupDRO on CelebA dataset.

Method	Learning rate	Weight decay	Worst	Average	Early-stopping epoch*
GroupDRO	1e-5	1e-1	$86.3_{\pm 1.1}$	$92.9_{\pm 0.3}$	$23.7_{\pm 6.8}$
PDE	1e-2	1e-4	$\mathbf{91.0}_{\pm 0.4}$	$92.0_{\pm 0.6}$	$\mathbf{0.7}_{\pm 0.3}$

Table 10: Training efficiency of PDE and GroupDRO on CivilComments-WILDS dataset.

Method	Learning rate	Weight decay	Worst	Average	Early-stopping epoch*
GroupDRO	1e-5	1e-2	$69.4_{\pm 0.9}$	$89.6_{\pm 0.5}$	$3.3_{\pm 2.1}$
PDE	1e-5	1e-2	$\mathbf{71.5}_{\pm 0.5}$	$86.3_{\pm 1.7}$	$\mathbf{2.1}_{\pm 1.1}$

Data Augmentation. Additionally, the increased training speed of our method facilitates the usage of techniques such as data augmentation. While data augmentation is a common practice for improving model generalization, DRO approaches have not incorporated it into their methods. We hypothesize that this omission stems from the slower training process. Data augmentation introduces random noise to the training data, which complicates convergence during training when using a very small learning rate. As illustrated in Table 11, data augmentation leads to slightly worse performance for GroupDRO. In contrast, our method effectively benefits from data augmentation.

Table 11: The effect of data augmentation on GroupDRO and PDE on Waterbirds dataset. We report the worst-group and average accuracy.

	GroupDRO		PDE	
Method	Worst	Avg	Worst	Avg
W/o data aug	86.7	93.2	88.9	89.5
W/ data aug	85.7	96.6	**90.3**	**92.4**

D Proof Preliminaries

Notation. In this paper, we use lowercase letters, lowercase boldface letters, and uppercase boldface letters to respectively denote scalars (a), vectors ($\mathbf{v}$), and matrices ($\mathbf{W}$). We use sgn to denote the sign function.For a vector $\mathbf{v}$, we use $\|\mathbf{v}\|_2$ to denote its Euclidean norm. Given two sequences $\{x_n\}$ and $\{y_n\}$, we denote $x_n = \mathcal{O}(y_n)$ if $|x_n| \leq C_1|y_n|$ for some absolute positive constant C_1, $x_n = \Omega(y_n)$ if $|x_n| \geq C_2|y_n|$ for some absolute positive constant C_2, and $x_n = \Theta(y_n)$ if $C_3|y_n| \leq |x_n| \leq C_4|y_n|$ for some absolute constants $C_3, C_4 > 0$. We use $\widetilde{\mathcal{O}}(\cdot)$ to hide logarithmic factors of d in $\mathcal{O}(\cdot)$.

Before we go into the analysis, we first consider the following gradient,

$$\nabla_{\mathbf{w}_j}\mathcal{L}(\mathbf{W}^{(t)}) = -\frac{1}{N}\sum_{i=1}^{N}\frac{\exp(-y_i f(\mathbf{x}_i;\mathbf{W}^{(t)}))}{1+\exp(-y_i f(\mathbf{x}_i;\mathbf{W}^{(t)}))}\cdot y_i f'(\mathbf{x}_i;\mathbf{W}^{(t)}). \tag{D.1}$$

Let's denote the derivative of a data example i at iteration t to be

$$\ell_i^{(t)} = \frac{\exp(-y_i f(\mathbf{x}_i;\mathbf{W}^{(t)}))}{1+\exp(-y_i f(\mathbf{x}_i;\mathbf{W}^{(t)}))} = \text{sigmoid}(-y_i f(\mathbf{x}_i;\mathbf{W}^{(t)})). \tag{D.2}$$

Lemma D.1. (Gradient) Let the loss function $\mathcal{L}$ be as defined in (2.1). For $t \geq 0$ and $j \in [J]$, the gradient of the loss $\mathcal{L}$ with regard to neuron $\mathbf{w}_j$ is

$$\nabla_{\mathbf{w}_j}\mathcal{L}(\mathbf{W}^{(t)}) = -\frac{3}{N}\bigg(\beta_c^3\sum_{i=1}^{N}\ell_i^{(t)}\langle\mathbf{w}_j,\mathbf{v}_c\rangle^2\mathbf{v}_c + \sum_{i=1}^{N}\ell_i^{(t)}y_i\langle\mathbf{w}_j,\boldsymbol{\xi}_i\rangle^2\boldsymbol{\xi}_i +$$

$$\Big(\sum_{i \in S_1} \ell_i^{(t)} - \sum_{i \in S_2} \ell_i^{(t)} \Big) \cdot \beta_s^3 \langle \mathbf{w}_j, \mathbf{v}_s \rangle^2 \mathbf{v}_s \Big).$$

Proof. We have the following gradient

$$\nabla_{\mathbf{w}_j} \mathcal{L}(\mathbf{W}^{(t)}) = -\frac{1}{N} \sum_{i=1}^{N} \frac{\exp(-y_i f(\mathbf{x}_i; \mathbf{W}^{(t)}))}{1 + \exp(-y_i f(\mathbf{x}_i; \mathbf{W}^{(t)}))} \cdot y_i f'(\mathbf{x}_i; \mathbf{W}^{(t)}). \tag{D.3}$$

And let's denote the derivative of a data example i at iteration t to be

$$\ell_i^{(t)} = \frac{\exp(-y_i f(\mathbf{x}_i; \mathbf{W}^{(t)}))}{1 + \exp(-y_i f(\mathbf{x}_i; \mathbf{W}^{(t)}))} = \text{sigmoid}(-y_i f(\mathbf{x}_i; \mathbf{W}^{(t)})). \tag{D.4}$$

Then, we can further write the gradient as

$$\begin{aligned}
\nabla_{\mathbf{w}_j} \mathcal{L}(\mathbf{W}^{(t)}) &= -\frac{3}{N} \sum_{i=1}^{N} \ell_i^{(t)} y_i \sum_{p=1}^{P} \langle \mathbf{w}_j, \mathbf{x}^{(p)} \rangle^2 \cdot \mathbf{x}^{(p)} \\
&= -\frac{3}{N} \sum_{i=1}^{N} \ell_i^{(t)} y_i \Big(\langle \mathbf{w}_j, \beta_c y_i \mathbf{v}_c \rangle^2 \beta_c y_i \mathbf{v}_c + \langle \mathbf{w}_j, \beta_s a_i \mathbf{v}_s \rangle^2 \beta_s a_i \mathbf{v}_s + \langle \mathbf{w}_j, \boldsymbol{\xi}_i \rangle^2 \boldsymbol{\xi}_i \Big) \\
&= -\frac{3}{N} \sum_{i=1}^{N} \ell_i^{(t)} \Big(\beta_c^3 \langle \mathbf{w}_j, \mathbf{v}_c \rangle^2 \mathbf{v}_c + \beta_s^3 y_i a_i \langle \mathbf{w}_j, \mathbf{v}_s \rangle^2 \mathbf{v}_s + y_i \langle \mathbf{w}_j, \boldsymbol{\xi}_i \rangle^2 \boldsymbol{\xi}_i \Big) \\
&= -\frac{3}{N} \Bigg(\sum_{i=1}^{N} \ell_i^{(t)} \Big(\beta_c^3 \langle \mathbf{w}_j, \mathbf{v}_c \rangle^2 \mathbf{v}_c + y_i \langle \mathbf{w}_j, \boldsymbol{\xi}_i \rangle^2 \boldsymbol{\xi}_i \Big) \\
&\qquad + \Big(\sum_{i \in S_1} \ell_i^{(t)} - \sum_{i \in S_2} \ell_i^{(t)} \Big) \beta_s^3 \langle \mathbf{w}_j, \mathbf{v}_s \rangle^2 \mathbf{v}_s \Bigg),
\end{aligned}$$

where the last equality holds due to that for $i \in S_1$ we have $a_i = y_i$ and for $i \in S_2$ we have $a_i = -y_i$. □

With the gradient, we have the following:

Core feature gradient. The projection of the gradient on $\mathbf{v}_c$ is then

$$\langle \nabla_{\mathbf{w}_j} \mathcal{L}(\mathbf{W}^{(t)}), \mathbf{v}_c \rangle = -\frac{3\beta_c^3}{N} \sum_{i=1}^{N} \ell_i^{(t)} \langle \mathbf{w}_j, \mathbf{v}_c \rangle^2. \tag{D.5}$$

Spurious feature gradient. The projection of the gradient on $\mathbf{v}_s$ is

$$\langle \nabla_{\mathbf{w}_j} \mathcal{L}(\mathbf{W}^{(t)}), \mathbf{v}_s \rangle = -\frac{3\beta_s^3}{N} \Big(\sum_{i \in S_1} \ell_i^{(t)} - \sum_{i \in S_2} \ell_i^{(t)} \Big) \cdot \langle \mathbf{w}_j, \mathbf{v}_s \rangle^2. \tag{D.6}$$

Noise gradient. The projection of the gradient on $\boldsymbol{\xi}_i$ is

$$\langle \nabla_{\mathbf{w}_j} \mathcal{L}(\mathbf{W}^{(t)}), \boldsymbol{\xi}_i \rangle = -\frac{3y_i}{N} \sum_{i=1}^{N} \ell_i^{(t)} \langle \mathbf{w}_j, \boldsymbol{\xi}_i \rangle^2 \|\boldsymbol{\xi}_i\|_2^2. \tag{D.7}$$

Derivative of data example i. $\ell_i^{(t)}$ can be rewritten as

$$\begin{aligned}
\ell_i^{(t)} &= \text{sigmoid}\big(-y_i f(\mathbf{x}_i; \mathbf{W}^{(t)}) \big) \\
&= \text{sigmoid}\Big(\sum_{j=1}^{J} -\beta_c^3 \langle \mathbf{w}_j, \mathbf{v}_c \rangle^3 - y_i a_i \beta_s^3 \langle \mathbf{w}_j, \mathbf{v}_s \rangle^3 - y_i \langle \mathbf{w}_j, \boldsymbol{\xi}_i \rangle^3 \Big).
\end{aligned} \tag{D.8}$$

Note that $0 < \ell_i^{(t)} < 1$ due to the property of the sigmoid function. Furthermore, we similarly consider that the sum of the sigmoid terms for all time steps is bounded up to a logarithmic dependence (Chen et al., 2022). And the sigmoid term is considered small for a κ such that

$$\sum_{t=0}^{T} \frac{1}{1+\exp(\kappa)} \leq \widetilde{O}(1),$$

which implies $\kappa \geq \widetilde{\Omega}(1)$.

E Proof of Theorem 2.2

In this section, we present the detailed proofs that build up to Theorem 2.2. We begin by considering the update for spurious feature and core feature.

Lemma E.1 (Spurious feature update.)**.** For all $t \geq 0$ and $j \in [J]$, the spurious feature update is

$$\langle \mathbf{w}_j^{(t+1)}, \mathbf{v}_s \rangle = \langle \mathbf{w}_j^{(t)}, \mathbf{v}_s \rangle + \frac{3\eta\beta_s^3}{N}\Big(\sum_{i\in S_1} \ell_i^{(t)} - \sum_{i\in S_2} \ell_i^{(t)}\Big)\langle \mathbf{w}_j^{(t)}, \mathbf{v}_s \rangle^2,$$

which gives

$$\begin{aligned}\widetilde{\Theta}(\eta)\beta_s^3\Big(\widehat{\alpha} g_1(t) - \sum_{i\in S_2} \ell_i^{(t)}/N\Big)\langle \mathbf{w}_j^{(t)}, \mathbf{v}_s \rangle^2 &\leq \langle \mathbf{w}_j^{(t+1)}, \mathbf{v}_s \rangle - \langle \mathbf{w}_j^{(t)}, \mathbf{v}_s \rangle \\ &\leq \widetilde{\Theta}(\eta)\beta_s^3 \cdot \widehat{\alpha} g_1(t) \cdot \langle \mathbf{w}_j^{(t)}, \mathbf{v}_s \rangle^2,\end{aligned}$$

where $g_1(t) = \text{sigmoid}\big(-\sum_{j\in[J]}(\beta_c^3\langle \mathbf{w}_j^{(t)}, \mathbf{v}_c\rangle^3 + \beta_s^3\langle \mathbf{w}_j^{(t)}, \mathbf{v}_s\rangle^3)\big)$.

Proof. The spurious feature update is obtained by using the gradient update of $\mathbf{W}^{(t)}$ and plugging in (D.6):

$$\begin{aligned}\langle \mathbf{w}_j^{(t+1)}, \mathbf{v}_s \rangle &= \langle \mathbf{w}_j^{(t)} - \eta\nabla_{\mathbf{w}_j}\mathcal{L}(\mathbf{W}^{(t)}), \mathbf{v}_s \rangle \\ &= \langle \mathbf{w}_j^{(t)}, \mathbf{v}_s \rangle + \frac{3\eta\beta_s^3}{N}\Big(\sum_{i\in S_1} \ell_i^{(t)} - \sum_{i\in S_2} \ell_i^{(t)}\Big)\langle \mathbf{w}_j^{(t)}, \mathbf{v}_s \rangle^2.\end{aligned}$$

We first prove for the upper bound. Consider the following,

$$\begin{aligned}\langle \mathbf{w}_j^{(t+1)}, \mathbf{v}_s \rangle &= \langle \mathbf{w}_j^{(t)}, \mathbf{v}_s \rangle + \frac{3\eta\beta_s^3}{N}\Big(\sum_{i\in S_1} \ell_i^{(t)} - \sum_{i\in S_2} \ell_i^{(t)}\Big)\langle \mathbf{w}_j^{(t)}, \mathbf{v}_s \rangle^2 \\ &\leq \langle \mathbf{w}_j^{(t)}, \mathbf{v}_s \rangle + \frac{3\eta\beta_s^3}{N}\Big(\sum_{i\in S_1} \ell_i^{(t)}\Big)\langle \mathbf{w}_j^{(t)}, \mathbf{v}_s \rangle^2 \\ &\leq \langle \mathbf{w}_j^{(t)}, \mathbf{v}_s \rangle + \widetilde{\Theta}(\eta)\beta_s^3 \cdot \frac{\sum_{i\in S_1} g_1(t)}{N} \cdot \langle \mathbf{w}_j^{(t)}, \mathbf{v}_s \rangle^2 \\ &= \langle \mathbf{w}_j^{(t)}, \mathbf{v}_s \rangle + \widetilde{\Theta}(\eta)\beta_s^3\widehat{\alpha} \cdot g_1(t) \cdot \langle \mathbf{w}_j^{(t)}, \mathbf{v}_s \rangle^2,\end{aligned}$$

where the first inequality holds due to $0 < \ell_i^{(t)} < 1$, the second inequality holds due to Lemma G.4, and the last equality holds due to $|S_1|/N = \widehat{\alpha}$. Then, for the lower bound, we consider the same bound for $i \in S_1$ in Lemma G.4 and obtain

$$\begin{aligned}\langle \mathbf{w}_j^{(t+1)}, \mathbf{v}_s \rangle &= \langle \mathbf{w}_j^{(t)}, \mathbf{v}_s \rangle + \frac{3\eta\beta_s^3}{N}\Big(\sum_{i\in S_1} \ell_i^{(t)} - \sum_{i\in S_2} \ell_i^{(t)}\Big)\langle \mathbf{w}_j^{(t)}, \mathbf{v}_s \rangle^2 \\ &\geq \langle \mathbf{w}_j^{(t)}, \mathbf{v}_s \rangle + \widetilde{\Theta}(\eta)\beta_s^3\Big(\widehat{\alpha} \cdot g_1(t) - \sum_{i\in S_2} \ell_i^{(t)}/N\Big)\langle \mathbf{w}_j^{(t)}, \mathbf{v}_s \rangle^2.\end{aligned}$$

□

Similarly, we have the update for core feature as below.

Lemma E.2 (Core feature update)**.** For all $t \geq 0$ and $j \in [J]$, the core feature update is

$$\langle \mathbf{w}_j^{(t+1)}, \mathbf{v}_c \rangle = \langle \mathbf{w}_j^{(t)}, \mathbf{v}_c \rangle + \frac{3\eta\beta_c^3}{N}\Big(\sum_{i=1}^N \ell_i^{(t)}\Big)\langle \mathbf{w}_j^{(t)}, \mathbf{v}_c \rangle^2,$$

which gives

$$\begin{aligned}\widetilde{\Theta}(\eta)\beta_c^3 \cdot \widehat{\alpha} g_1(t) \cdot \langle \mathbf{w}_j^{(t)}, \mathbf{v}_c \rangle^2 &\leq \langle \mathbf{w}_j^{(t+1)}, \mathbf{v}_c \rangle - \langle \mathbf{w}_j^{(t)}, \mathbf{v}_c \rangle \\ &\leq \widetilde{\Theta}(\eta)\beta_c^3 \cdot \Big(\widehat{\alpha} g_1(t) + \sum_{i \in S_2} \ell_i^{(t)}/N\Big) \cdot \langle \mathbf{w}_j^{(t)}, \mathbf{v}_c \rangle^2.\end{aligned}$$

Proof. The core feature update is obtained by using the gradient update of $\mathbf{W}^{(t)}$ and plugging in (D.5):

$$\begin{aligned}\langle \mathbf{w}_j^{(t+1)}, \mathbf{v}_c \rangle &= \langle \mathbf{w}_j^{(t)} - \eta \nabla_{\mathbf{w}_j} \mathcal{L}(\mathbf{W}^{(t)}), \mathbf{v}_c \rangle \\ &= \langle \mathbf{w}_j^{(t)}, \mathbf{v}_c \rangle + \frac{3\eta\beta_c^3}{N}\Big(\sum_{i=1}^N \ell_i^{(t)}\Big)\langle \mathbf{w}_j^{(t)}, \mathbf{v}_c \rangle^2.\end{aligned}$$

We prove for the lower bound,

$$\begin{aligned}\langle \mathbf{w}_j^{(t+1)}, \mathbf{v}_c \rangle &= \langle \mathbf{w}_j^{(t)}, \mathbf{v}_c \rangle + \frac{3\eta\beta_c^3}{N}\Big(\sum_{i=1}^N \ell_i^{(t)}\Big)\langle \mathbf{w}_j^{(t)}, \mathbf{v}_c \rangle^2 \\ &\geq \langle \mathbf{w}_j^{(t)}, \mathbf{v}_c \rangle + \frac{3\eta\beta_c^3}{N}\Big(\sum_{i \in S_1} \ell_i^{(t)}\Big)\langle \mathbf{w}_j^{(t)}, \mathbf{v}_c \rangle^2 \\ &\geq \langle \mathbf{w}_j^{(t)}, \mathbf{v}_c \rangle + \widetilde{\Theta}(\eta)\beta_c^3 \widehat{\alpha} g_1(t) \cdot \langle \mathbf{w}_j^{(t)}, \mathbf{v}_c \rangle^2,\end{aligned}$$

where the first inequality holds due to $0 < \ell_i^{(t)} < 1$ and the second inequality holds due to Lemma G.4. And for the upper bound. we similarly have

$$\begin{aligned}\langle \mathbf{w}_j^{(t+1)}, \mathbf{v}_c \rangle &= \langle \mathbf{w}_j^{(t)}, \mathbf{v}_c \rangle + \frac{3\eta\beta_c^3}{N}\Big(\sum_{i=1}^N \ell_i^{(t)}\Big)\langle \mathbf{w}_j^{(t)}, \mathbf{v}_c \rangle^2 \\ &\leq \langle \mathbf{w}_j^{(t)}, \mathbf{v}_c \rangle + \widetilde{\Theta}(\eta)\beta_c^3 \cdot \Big(\widehat{\alpha} g_1(t) + \sum_{i \in S_2} \ell_i^{(t)}\Big) \cdot \langle \mathbf{w}_j^{(t)}, \mathbf{v}_c \rangle^2.\end{aligned}$$

□

Note that $\langle \mathbf{w}_j^{(t+1)}, \mathbf{v}_c \rangle$ is non-decreasing from the lower bound of Lemma E.2. As $\mathbf{w}_j^{(0)} \sim \mathcal{N}(0, \sigma_0^2 \mathbf{I}_d)$ are initialized with small σ_0, the sigmoid terms $\ell_i^{(t)}$ are large in the initial iterations. And while $l_i^{(t)}$ remains large for $i \in S_1$, we have $g_1(t) = \Theta(1)$ as similar as in Jelassi & Li (2022). Therefore, $\langle \mathbf{w}_j^{(t+1)}, \mathbf{v}_s \rangle$ is also non-decreasing since $\widehat{\alpha} \cdot \Theta(1) - \sum_{i \in S_2} l_i^{(t)}/N \geq 2\widehat{\alpha} - 1 > 0$ for $l_i^{(t)} < 1$ and $\widehat{\alpha} > 1/2$. Eventually, $g_1(t)$ becomes small at a time $T_0 > 0$. We now consider a simplified version of the above lemma in this early training stage.

Lemma E.3 (Spurious feature update in early iterations)**.** Let $T_0 > 0$ be such that $\max_{j \in [J]} \langle \mathbf{w}_j^{(T_0)}, \mathbf{v}_s \rangle \geq \widetilde{\Omega}(1/\beta_s)$. For $t \in [0, T_0]$, the spurious feature update has the following bound

$$\widetilde{\Theta}(\eta)\beta_s^3(2\widehat{\alpha} - 1) \cdot \langle \mathbf{w}_j^{(t)}, \mathbf{v}_s \rangle^2 \leq \langle \mathbf{w}_j^{(t+1)}, \mathbf{v}_s \rangle - \langle \mathbf{w}_j^{(t)}, \mathbf{v}_s \rangle \leq \widetilde{\Theta}(\eta)\beta_s^3 \widehat{\alpha} \cdot \langle \mathbf{w}_j^{(t)}, \mathbf{v}_s \rangle^2.$$

Proof. Let $T_0 > 0$ be such that either $\max_{j \in [J]} \langle \mathbf{w}_j^{(T_0)}, \mathbf{v}_s \rangle \geq \widetilde{\Omega}(1/\beta_s)$ or $\max_{j \in [J]} \langle \mathbf{w}_j^{(T_0)}, \mathbf{v}_c \rangle \geq \widetilde{\Omega}(1/\beta_c)$. We will show later that the first condition will be first met and we have $\langle \mathbf{w}_j^{(t)}, \mathbf{v}_c \rangle \leq \widetilde{\Omega}(1/\beta_c)$ for all $j \in [J]$ and $t \in [0, T_0]$.

Recall that $g_1(t) = \text{sigmoid}\big(-\sum_{j\in[J]}(\beta_c^3\langle \mathbf{w}_j^{(t)}, \mathbf{v}_c\rangle^3 + \beta_s^3\langle \mathbf{w}_j^{(t)}, \mathbf{v}_s\rangle^3)\big)$. Then, for $t \in [0, T_0]$, we have

$$\begin{aligned} g_1(t) &= \frac{1}{1+\exp\big(\sum_{j\in[J]}(\beta_c^3\langle \mathbf{w}_j^{(t)}, \mathbf{v}_c\rangle^3 + \beta_s^3\langle \mathbf{w}_j^{(t)}, \mathbf{v}_s\rangle^3)\big)} \\ &\geq \frac{1}{1+\exp\big(\kappa+\kappa\big)} \\ &= \frac{1}{1+\exp\big(\widetilde{\Omega}(1)\big)}, \end{aligned}$$

where the first inequality holds due to $\langle \mathbf{w}_s^{(t)}, \mathbf{v}_s\rangle \leq \kappa/(J^{1/3}\beta_s)$ and $\langle \mathbf{w}_s^{(t)}, \mathbf{v}_c\rangle \leq \kappa/(J^{1/3}\beta_c)$ for $t \in [0, T_0]$. Therefore, similar to Jelassi & Li (2022), we have $g_1(t) = \Theta(1)$ in the early iterations. Moreover, as $0 < \ell_i^{(t)} < 1$, we have $\sum_{i\in S_2} \ell_i^{(t)}/N < 1 - \widehat{\alpha}$. This implies the result in Lemma E.1 as

$$\widetilde{\Theta}(\eta)\beta_s^3(2\widehat{\alpha}-1)\langle \mathbf{w}_j^{(t)}, \mathbf{v}_s\rangle^2 \leq \langle \mathbf{w}_j^{(t+1)}, \mathbf{v}_s\rangle - \langle \mathbf{w}_j^{(t)}, \mathbf{v}_s\rangle \leq \widetilde{\Theta}(\eta)\beta_s^3\widehat{\alpha}\langle \mathbf{w}_j^{(t)}, \mathbf{v}_s\rangle^2.$$

□

And similarly for core feature, we have

Lemma E.4 (Core feature update in early iterations)**.** Let $T_0 > 0$ be such that $\max_{j\in[J]}\langle \mathbf{w}_j^{(T_0)}, \mathbf{v}_s\rangle \geq \widetilde{\Omega}(1/\beta_s)$. For $t \in [0, T_0]$, the core feature update has the following bound

$$\widetilde{\Theta}(\eta)\beta_c^3\widehat{\alpha}\cdot\langle \mathbf{w}_j^{(t)}, \mathbf{v}_c\rangle^2 \leq \langle \mathbf{w}_j^{(t+1)}, \mathbf{v}_c\rangle - \langle \mathbf{w}_j^{(t)}, \mathbf{v}_c\rangle \leq \widetilde{\Theta}(\eta)\beta_c^3\cdot\langle \mathbf{w}_j^{(t)}, \mathbf{v}_c\rangle^2.$$

Proof. Let $T_0 > 0$ be such that either $\max_{j\in[J]}\langle \mathbf{w}_j^{(T_0)}, \mathbf{v}_s\rangle \geq \widetilde{\Omega}(1/\beta_s)$ or $\max_{j\in[J]}\langle \mathbf{w}_j^{(T_0)}, \mathbf{v}_c\rangle \geq \widetilde{\Omega}(1/\beta_c)$. Again, with $g_1(t) = \Theta(1)$ and $\sum_{i\in S_2} \ell_i^{(t)}/N < 1 - \widehat{\alpha}$ as shown in Lemma E.3, we can imply the result in Lemma E.2 as

$$\widetilde{\Theta}(\eta)\beta_c^3\widehat{\alpha}\cdot\langle \mathbf{w}_j^{(t)}, \mathbf{v}_c\rangle^2 \leq \langle \mathbf{w}_j^{(t+1)}, \mathbf{v}_c\rangle - \langle \mathbf{w}_j^{(t)}, \mathbf{v}_c\rangle \leq \widetilde{\Theta}(\eta)\beta_c^3\cdot\langle \mathbf{w}_j^{(t)}, \mathbf{v}_c\rangle^2,$$

which completes the proof. □

With the updates of spurious and core feature in the early iterations, we can now show with the following lemma that GD will learn the spurious feature very quickly while hardly learns the core feature.

Lemma E.5. Let T_0 be the iteration number that $\max_{j\in[J]}\langle \mathbf{w}_j^{(t)}, \mathbf{v}_s\rangle$ reaches $\widetilde{\Omega}(1/\beta_s) = \widetilde{\Theta}(1)$. Then, we have for all $t \leq T_0$, it holds that $\max_{j\in[J]}\langle \mathbf{w}_j^{(t)}, \mathbf{v}_c\rangle = \widetilde{O}(\sigma_0)$.

Proof. Consider the following from Lemma E.3 and Lemma E.4,

$$\begin{aligned} &\langle \mathbf{w}_j^{(t+1)}, \mathbf{v}_c\rangle - \langle \mathbf{w}_j^{(t)}, \mathbf{v}_c\rangle \leq \widetilde{\Theta}(\eta)\beta_c^3\cdot\langle \mathbf{w}_j^{(t)}, \mathbf{v}_c\rangle^2 \\ &\langle \mathbf{w}_j^{(t+1)}, \mathbf{v}_s\rangle - \langle \mathbf{w}_j^{(t)}, \mathbf{v}_s\rangle \geq \widetilde{\Theta}(\eta)\beta_s^3(2\widehat{\alpha}-1)\langle \mathbf{w}_j^{(t)}, \mathbf{v}_s\rangle^2. \end{aligned}$$

Recall that we initialize the weights as $\mathbf{w}_j^{(0)} \sim \mathcal{N}(\mathbf{0}, \sigma_0^2)$. We have $\langle \mathbf{w}_j^{(0)}, \mathbf{v}_c\rangle \sim \mathcal{N}(0, \sigma_0^2)$ and $\langle \mathbf{w}_j^{(0)}, \mathbf{v}_s\rangle \sim \mathcal{N}(0, \sigma_0^2)$. For the weights have small initialization with $\sigma_0 = \text{polylog}(d)/d$, we have $O(\langle \mathbf{w}_j^{(0)}, \mathbf{v}_c\rangle) = O(\langle \mathbf{w}_j^{(0)}, \mathbf{v}_s\rangle)$. Therefore, for $\beta_c^3 = o(1)$ and $\beta_s^3(2\widehat{\alpha}-1) = \Theta(1)$, we call Lemma G.1 and get

$$\langle \mathbf{w}_j^{(T_0)}, \mathbf{v}_c\rangle \leq O(\langle \mathbf{w}_j^{(0)}, \mathbf{v}_s\rangle) = \widetilde{O}(\sigma_o)$$

for all $j \in [J]$. □

Given the above lemma, we can conclude that the condition $\max_{j\in[J]}\langle \mathbf{w}_j^{(T_0)}, \mathbf{v}_s\rangle \geq \widetilde{\Omega}(1/\beta_s)$ will be first met. And therefore, T_0 is such that $\max_{j\in[J]}\langle \mathbf{w}_j^{(T_0)}, \mathbf{v}_s\rangle \geq \widetilde{\Omega}(1/\beta_s)$.

Theorem E.6 (Restatement of Theorem 2.2)**.** Consider the training dataset $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ that follows the distribution in Definition 2.1. Consider the two-layer nonlinear CNN model as in (2.3) initialized with $\mathbf{W}^{(0)} \sim \mathcal{N}(0, \sigma_0^2)$. After training with GD in (2.2) for $T_0 = \widetilde{\Theta}\big(1/(\eta\beta_s^3\sigma_0)\big)$ iterations, for all $j \in [J]$ and $t \in [0, T_0)$, we have

$$\widetilde{\Theta}(\eta)\beta_s^3(2\widehat{\alpha}-1)\cdot\langle\mathbf{w}_j^{(t)},\mathbf{v}_s\rangle^2 \le \langle\mathbf{w}_j^{(t+1)},\mathbf{v}_s\rangle - \langle\mathbf{w}_j^{(t)},\mathbf{v}_s\rangle \le \widetilde{\Theta}(\eta)\beta_s^3\widehat{\alpha}\cdot\langle\mathbf{w}_j^{(t)},\mathbf{v}_s\rangle^2, \tag{E.1}$$

$$\widetilde{\Theta}(\eta)\beta_c^3\widehat{\alpha}\cdot\langle\mathbf{w}_j^{(t)},\mathbf{v}_c\rangle^2 \le \langle\mathbf{w}_j^{(t+1)},\mathbf{v}_c\rangle - \langle\mathbf{w}_j^{(t)},\mathbf{v}_c\rangle \le \widetilde{\Theta}(\eta)\beta_c^3\cdot\langle\mathbf{w}_j^{(t)},\mathbf{v}_c\rangle^2. \tag{E.2}$$

After training for T_0 iterations, with high probability, the learned weight has the following properties: (1) it learns the spurious feature $\mathbf{v}_s$: $\max_{j\in[J]}\langle\mathbf{w}_j^{(T)},\mathbf{v}_s\rangle \ge \widetilde{\Omega}(1/\beta_s)$; (2) it does not learn the core feature $\mathbf{v}_c$: $\max_{j\in[J]}\langle\mathbf{w}_j^{(T)},\mathbf{v}_c\rangle = \widetilde{\mathcal{O}}(\sigma_0)$.

Proof. The updates directly follow the results from Lemma E.1 and Lemma E.2. And the result for $\max_{j\in[J]}\langle\mathbf{w}_j^{(t)},\mathbf{v}_c\rangle$ follows Lemma E.5. It remains to calculate the time T_0. With Lemma G.2, we consider the sequence for $\max_{j\in[J]}\langle\mathbf{w}_j^{(t+1)},\mathbf{v}_s\rangle$, where by Lemma E.3,

$$\langle\mathbf{w}_j^{(t+1)},\mathbf{v}_s\rangle \le \langle\mathbf{w}_j^{(t)},\mathbf{v}_s\rangle + \widetilde{\Theta}(\eta)\beta_s^3\widehat{\alpha}\cdot\langle\mathbf{w}_j^{(t)},\mathbf{v}_s\rangle^2,$$
$$\langle\mathbf{w}_j^{(t+1)},\mathbf{v}_s\rangle \ge \langle\mathbf{w}_j^{(t)},\mathbf{v}_s\rangle + \widetilde{\Theta}(\eta)\beta_s^3(2\widehat{\alpha}-1)\cdot\langle\mathbf{w}_j^{(t)},\mathbf{v}_s\rangle^2.$$

As $\langle\mathbf{w}_j^{(t)},\mathbf{v}_s\rangle$ is non-decreasing in early iterations and with high probability, there exist an index j such that $\langle\mathbf{w}_j^{(0)},\mathbf{v}_s\rangle \ge 0$. Among all the possible indices $i \in [J]$ that are initialized to have posive inner product with $\mathbf{v}_s$, we focus on the max index $r = \operatorname{argmax}_{j\in[J]}\langle\mathbf{w}_j^{(0)},\mathbf{v}_s\rangle$. Then with $v = \widetilde{\Theta}(1/\beta_s)$ in Lemma G.2, we will have T_0 as

$$T_0 = \frac{\widetilde{\Theta}(1)}{\eta\alpha^3\sigma_0} + \frac{\widetilde{\Theta}(1)\widehat{\alpha}}{2\widehat{\alpha}-1}\left\lceil\frac{-\log\big(\sigma_0\beta_s\big)}{\log(2)}\right\rceil.$$

□

F Proof of Lemma 3.1

Lemma F.1 (Restatement of Lemma 3.1)**.** Given the balanced training dataset $S^0 = \{(\mathbf{x}_i, y_i, a_i)\}_{i=1}^{N_0}$ with $\widehat{\alpha} = 1/2$ as in Definition 2.1 and CNN as in (2.3). The gradient on $\mathbf{v}_s$ will be 0 from the beginning of training.

Proof. With Lemma D.1, the projection of the gradient on $\mathbf{v}_s$ in the initial iteration ($t < T_0$) is

$$\begin{aligned}\langle\nabla_{\mathbf{w}_j}\mathcal{L}(\mathbf{W}^{(t)}),\mathbf{v}_s\rangle &= -\frac{3\beta_s^3}{N}\Big(\sum_{i\in S_1}\ell_i^{(t)} - \sum_{i\in S_2}\ell_i^{(t)}\Big)\cdot\langle\mathbf{w}_j,\mathbf{v}_s\rangle^2\\ &= \Theta\left(\frac{\beta_s^3}{N}\right)\Big(|S_1| - |S_2|\Big)\\ &= 0,\end{aligned}$$

where the first equality is due to $\ell_i^{(t)} = \Theta(1)$ in the initial iterations and the second equality is due to $\widehat{\alpha} = 0.5$. □

G Auxiliary Lemmas

Lemma G.1 (Lemma C.20, Allen-Zhu & Li 2020)**.** Let $\{x_t, y_t\}_{t=1,..}$ be two positive sequences that satisfy

$$x_{t+1} \ge x_t + \eta\cdot Ax_t^2,$$
$$y_{t+1} \le y_t + \eta\cdot By_t^2,$$

for some $A = \Theta(1)$ and $B = o(1)$. Suppose $y_0 = O(x_0)$ and $\eta < O(x_0)$, and for all $C \in [X_0, O(1)]$, let T_x be the first iteration such that $x_t \ge C$. Then, we have $T_x\eta = \Theta(x_0^{-1})$ and

$$y_{T_x} \le O(x_0).$$

Lemma G.2 (Lemma K.15, Jelassi & Li 2022)**.** Let $\{z_t\}_{t=0}^T$ be a positive sequence defined by the following recursions

$$z_{t+1} \geq z_t + m(z_t)^2,$$
$$z_{t+1} \leq z_t + M(z_t)^2,$$

where $z_0 > 0$ is the initialization and $m, M > 0$ are some constants. Let $v > 0$ such that $z_0 \leq v$. Then, the time t_0 such that $z_t \geq v$ for all $t \geq t_0$ is

$$t_0 = \frac{3}{mz_0} + \frac{8M}{m}\left\lceil \frac{\log(v/z_0)}{\log(2)} \right\rceil.$$

We make the following assumptions for every $t \leq T$ as the same in (Jelassi & Li, 2022).

Lemma G.3 (Induction hypothesis D.1, Jelassi & Li 2022)**.** Throughout the training process using GD for $t \leq T$, we maintain that, for every $i \in S_1$ and $j \in [J]$,

$$|\langle \mathbf{w}_j^{(t)}, \boldsymbol{\xi}_i \rangle| \leq \widetilde{O}(\sigma_0 \sigma \sqrt{d}). \tag{G.1}$$

Lemma G.4. For $i \in S_1$, we have $\ell_i^{(t)} = \Theta(1) g_1(t)$, where

$$g_1(t) = \text{sigmoid}\big(- \sum_{j \in [J]} (\beta_c^3 \langle \mathbf{w}_j^{(t)}, \mathbf{v}_c \rangle^3 + \beta_s^3 \langle \mathbf{w}_j^{(t)}, \mathbf{v}_s \rangle^3) \big).$$

Proof. Given $i \in S_1$, we have from (D.8) that

$$\begin{aligned} \ell_i^{(t)} &= \text{sigmoid}\Bigg(\sum_{j=1}^J -\beta_c^3 \langle \mathbf{w}_j, \mathbf{v}_c \rangle^3 - \beta_s^3 \langle \mathbf{w}_j, \mathbf{v}_s \rangle^3 - y_i \langle \mathbf{w}_j, \boldsymbol{\xi}_i \rangle^3 \Bigg) \\ &= 1 \Bigg/ \Bigg(1 + \exp\Big(\sum_{j=1}^J \beta_c^3 \langle \mathbf{w}_j, \mathbf{v}_c \rangle^3 + \beta_s^3 \langle \mathbf{w}_j, \mathbf{v}_s \rangle^3 + y_i \langle \mathbf{w}_j, \boldsymbol{\xi}_i \rangle^3 \Big) \Bigg). \end{aligned} \tag{G.2}$$

Recall induction hypothesis G.3, we have the following for $i \in S_1$,

$$\begin{aligned} &|y_i \langle \mathbf{w}_j^{(t)}, \boldsymbol{\xi}_i \rangle| \leq \widetilde{O}(\sigma_0 \sigma \sqrt{d}) \\ &\iff -\widetilde{O}(\sigma_0 \sigma \sqrt{d}) \leq y_i \langle \mathbf{w}_j^{(t)}, \boldsymbol{\xi}_i \rangle \leq \widetilde{O}(\sigma_0 \sigma \sqrt{d}), \end{aligned} \tag{G.3}$$

where $|y_i| = 1$. Plug (G.3) back into (G.2), we get

$$e^{-\widetilde{O}(\sigma_0 \sigma \sqrt{d})^3} g_1(t) \leq \ell_i^{(t)} \leq e^{\widetilde{O}(\sigma_0 \sigma \sqrt{d})^3} g_1(t).$$

With our parameter setting, we have $\widetilde{O}(\sigma_0 \sigma \sqrt{d}) = \widetilde{O}(\sigma_0) = \widetilde{O}(\text{polylog}(d)/d)$. Therefore, $e^{\pm \widetilde{O}(\sigma_0 \sigma \sqrt{d})^3} = \Theta(1)$. □