# Identifying Spurious Biases Early in Training through the Lens of Simplicity Bias

**Yu Yang**
University of California, Los Angeles

**Eric Gan**
University of California, Los Angeles

**Gintare Karolina Dziugaite**
Google Deepmind

**Baharan Mirzasoleiman**
University of California, Los Angeles

## Abstract

Neural networks trained with (stochastic) gradient descent have an inductive bias towards learning simpler solutions. This makes them highly prone to learning *spurious correlations* in the training data, that may not hold at test time. In this work, we provide the first theoretical analysis of the effect of simplicity bias on learning spurious correlations. Notably, we show that examples with spurious features are *provably* separable based on the model's output *early in training*. We further illustrate that if spurious features have a small enough noise-to-signal ratio, the network's output on majority of examples is almost exclusively determined by the spurious features, leading to poor *worst-group* test accuracy. Finally, we propose SPARE, which identifies spurious correlations early in training, and utilizes importance sampling to alleviate their effect. Empirically, we demonstrate that SPARE outperforms state-of-the-art methods by up to 21.1% in worst-group accuracy, while being up to 12x faster. We also show that SPARE is a highly effective but lightweight method to *discover spurious correlations*. Code is available at https://github.com/BigML-CS-UCLA/SPARE.

## 1 INTRODUCTION

The *simplicity bias* of gradient-based training algorithms towards learning simpler solutions has been suggested as a key factor for the superior generalization performance of overparameterized neural networks (Hermann and Lampinen, 2020; Hu et al., 2020; Nakkiran et al., 2019; Neyshabur et al., 2014; Pezeshki et al., 2021; Shah et al., 2020). At the same time, it is conjectured to make neural networks vulnerable to learning *spurious correlations* frequently found in real-world datasets (Sagawa et al., 2019; Sohoni et al., 2020). Neural networks trained with gradient-based methods can exclusively rely on simple *spurious features* that are highly correlated with a class in the training data but are not predictive of the class in general, and remain invariant to the predictive but more complex *core features* (Shah et al., 2020). This results in a poor *worst-group test accuracy* on groups of examples where the spurious correlations do not hold (Shah et al., 2020; Teney et al., 2022). For example, in an image classification task, if the majority of images of a 'bird' appear on a 'sky' background, the classifier learns the sky instead of bird, and misclassifies birds that do not appear in the sky at test time.

An effective way to mitigate a spurious correlation and improve the worst-group test accuracy is to upweight examples that do not contain the spurious feature during training (Sagawa et al., 2019). However, inspecting all training examples to find such examples becomes prohibitive in real-world datasets. This has motivated a growing body of work on group inference: separating majority groups exhibiting spurious correlation with a class, from minority groups without the spurious correlation. Such methods first train a neural network with gradient methods to learn the spurious correlation. Then, they rely on model's misclassification (Liu et al., 2021), loss (Creager et al., 2021), or representations (Sohoni et al., 2020; Ahmed et al., 2020) at a certain point during training, as an indicative of minority examples. The time of group inference and how much to upweight the minority groups are heavily tuned based on a group-labeled validation data (Sohoni et al., 2020; Liu et al.,

Proceedings of the 27$^{\text{th}}$ International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238. Copyright 2024 by the author(s).

2021; Creager et al., 2021; Ahmed et al., 2020).

Despite their success on simple benchmark datasets, we show that such methods suffer from several issues: (1) they often misidentify minority examples as majority and mistakenly downweight them; then, (2) to counteract the spurious correlation they need to heavily upweight their small inferred minority group. This magnifies milder spurious correlations that may exist in the minority group (Li et al., 2023) and harms the performance; as (3) there is no theoretical guideline for finding the time of group inference and group weights, such methods rely on extensive hyperparameter tuning. This limits their applicability and scalability.

In this work, we make several theoretical and empirical contributions towards addressing the above issues. First, we prove that the simplicity bias of gradient descent can be leveraged to identify spurious correlations. We analyze a two-layer fully connected neural network trained with SGD, and leverage recent results showing its early-time learning dynamics can be mimicked by training a linear model on the inputs (Hu et al., 2020). We show that the contribution of a spurious feature to the network output in the initial training phase increases linearly with the amount of spurious correlation. Thus, minority and majority groups can be *provably* separated based on the model's output, *early in training*. This enables more accurate identification of minorities, and limits the range of group inference to the first few training epochs, without extensive hyperparameter tuning.

Next, we show that once the initial linear model converges, if the noise-to-signal ratio of a spurious feature is lower than that of the core feature in a class, the network will not learn the core features of the majority groups. This explains prior empirical observations (Shah et al., 2020), by revealing *when and why* neural networks trained with gradient almost exclusively rely on spurious features and remain invariant to the predictive but more complex core features. To the best of our knowledge, this is the first analysis of the effect of SGD's simplicity bias on learning spurious vs core features.

Finally, we propose an efficient and lightweight method, Spare (SePArate early and REsample), that clusters model's output early in training, and leverage importance sampling based on inverse cluster sizes to mitigate spurious correlations. This results in a superior worst-group accuracy on more challenging tasks, without increasing the training time, or requiring extensive hyperparameter tuning. Unlike existing methods, Spare can operate without a group-labeled validation data, which allows it to *discover unknown spurious correlations.*

Our extensive experiments confirm that Spare achieves up to 42.9% higher worst-group accuracy over state-of-the-art on most commonly used benchmarks, including CMNIST (Alain et al., 2015) (with multiple minority groups), Waterbirds (Sagawa et al., 2019), CelebA (Liu et al., 2015) and UrbanCars (Li et al., 2023) (with multiple spurious correlations) while being up to 12x faster. On CMNIST, Spare performs well across varying noise-to-signal ratios, whereas other state-of-the-art methods struggle. Applied to Restricted ImageNet, a dataset without known spurious correlations or group-labeled validation set available for hyperparameter tuning, identifies the spurious correlation much more effectively than the state-of-the-art group inference methods and improves the model's accuracy on minority groups by up to 23.2% higher than them after robust training.

## 2 RELATED WORK

**Mitigating Spurious Correlations With Group Inference.** To mitigate spurious correlations, group inference methods typically fall into three categories: (1) end-of-training clustering, (2) Environment Invariance Maximization, and (3) misclassification methods. GEORGE (Sohoni et al., 2020) utilizes end-of-training clustering by first training a model using Empirical Risk Minimization (ERM). The clustered feature representations are then used to train a robust model with Group Distributionally Robust Optimization (GDRO) (Sagawa et al., 2019). Methods use Environment Invariance Maximization, EIIL (Creager et al., 2021) and PGI (Ahmed et al., 2020), train an initial model using ERM and then partition the data to maximize the Invariant Risk Minimization (IRM) objective (Arjovsky and Bottou, 2017). Then, EIIL trains the robust model with GDRO, whereas PGI minimizes the KL divergence of softmaxed logits for same-class samples across groups. The misclassification approach, exemplified by JTT (Liu et al., 2021) trains an ERM model for some epochs and identifies misclassified examples. The training set is then upsampled with these examples, and a robust model is trained using ERM on this augmented set.

State-of-the-art methods commonly misidentify minority examples as the majority and struggle to mitigate spurious correlations. They heavily rely on a group-labeled validation for hyperparameter tuning, and often increase training time during group inference or robust training. In contrast, guided by theory, Spare can accurately separate groups with spurious features in the first few epochs, eliminating the need for extensive hyperparameter tuning and achieving better performance on minority groups without extending training time.

**Simplicity Bias.** Simplicity bias of SGD in learning simpler functions before complex ones (Hermann and Lampinen, 2020; Hu et al., 2020; Nakkiran et al., 2019;

Neyshabur et al., 2014; Pezeshki et al., 2021; Shah et al., 2020) is empirically observed in various network architectures, including MobileNetV2, ResNet50, and DenseNet121 (Sandler et al., 2018; He et al., 2016; Shah et al., 2020). Hu et al. (2020) formally proved that initial learning dynamics of a two-layer FC network can be mimicked by a linear model and extended this to multi-layer FC and convolutional networks. While simplicity bias helps overparameterized networks generalize well, it is also *conjectured* to favor simpler features over complex ones, even if they are less predictive (Shah et al., 2020; Teney et al., 2022). However, the exact notion of the simplicity of features and the mechanism by which they are learned remain poorly understood except in certain simplistic settings (Nagarajan et al., 2020; Shah et al., 2020). In this work, we build on Hu et al. (2020) to rigorously specify the conditions and mechanism of learning spurious features in a two-layer FC network.

## 3 PROBLEM FORMULATION

Let $\mathcal{D}=\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n \subset \mathbb{R}^d \times \mathbb{R}$ be $n$ training data with features $\boldsymbol{x}_i \in \mathbb{R}^d$, and labels $y_i \in \mathcal{C}=\{1,-1\}$. For simplicity, we consider binary classification with $\ell_2$ loss, but our analysis generalizes to multi-class classification with CE loss, and other model architectures, as we will confirm experimentally.

**Features & Groups.** We assume every class $c \in \mathcal{C}$ has a *core* feature $\boldsymbol{v}_c$, which is the invariant feature of the class that appears in both training and test sets. Besides, there is a set of *spurious* features $\mathcal{A}$ that are shared between classes and are present in both training and test sets, but may not have a spurious correlation with the labels at test time. For example, in the CMNIST dataset containing images of colored handwritten digits (Fig. 1), the digit is the core feature, and its color is the spurious feature. Assuming w.l.o.g. that all $\boldsymbol{v}_c, \boldsymbol{v}_s \in \mathbb{R}^d$ are orthogonal vectors, the feature vector of every example $\boldsymbol{x}_i$ in class $c$ can be written as $\boldsymbol{x}_i = \boldsymbol{v}_c + \boldsymbol{v}_s + \boldsymbol{\xi}_i$, where $\boldsymbol{v}_s \in \mathcal{A}$, and each $\boldsymbol{\xi}_i$ is a noise vector drawn i.i.d. from $\mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_\xi)$. Noise-to-signal (NSR) ratio of a feature is defined as its variance over magnitude, i.e., $R_\cdot = \sigma_\cdot/\|\boldsymbol{v}_\cdot\|$. Features with smaller NSR are simpler to learn. Training examples can be partitioned into groups $g_{c,s}$ based on the combinations of their core and spurious features $(\boldsymbol{v}_c, \boldsymbol{v}_s)$.

**Neural Network & Training.** We consider a two-layer FC neural network with $m$ hidden neurons:

$$f(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{z}) = \frac{1}{\sqrt{m}} \sum_{r=1}^{m} z_r \phi(\boldsymbol{w}_r^T \boldsymbol{x}/\sqrt{d}) = \frac{1}{\sqrt{m}} \boldsymbol{z}^T \phi(\boldsymbol{W}\boldsymbol{x}/\sqrt{d}),$$

where $\boldsymbol{x} \in \mathbb{R}^d$ is the input, $\boldsymbol{W} = [\boldsymbol{w}_1, \cdots, \boldsymbol{w}_m]^T \in \mathbb{R}^{m \times d}$ is the weight matrix in the first layer, and $\boldsymbol{z} = [z_1, \cdots, z_m]^T \in \mathbb{R}^m$ is the weight vector in the second layer. Here $\phi : \mathbb{R} \to \mathbb{R}$ is a smooth or piecewise linear activation function (including ReLU, Leaky ReLU, Erf, Tanh, Sigmoid, Softplus, etc.) that acts entry-wise on vectors or matrices. We consider the following $\ell_2$ training loss:

$$\mathcal{L}(\boldsymbol{W}, \boldsymbol{z}) = \frac{1}{2n} \sum_{i=1}^{n} (f(\boldsymbol{x}_i; \boldsymbol{W}, \boldsymbol{z}) - y_i)^2. \tag{1}$$

We train the network by applying gradient descent on the loss (1) starting from random initialization[1]:

$$\boldsymbol{W}_{t+1} = \boldsymbol{W}_t - \eta \nabla_{\boldsymbol{W}} \mathcal{L}(\boldsymbol{W}_t, \boldsymbol{z}_t), \tag{2}$$

$$\boldsymbol{z}_{t+1} = \boldsymbol{z}_t - \eta \nabla_{\boldsymbol{z}} \mathcal{L}(\boldsymbol{W}_t, \boldsymbol{z}_t), \tag{3}$$

**Worst-group Error.** We quantify the performance of the model based on its highest test error across groups $\mathcal{G} = \{g_{c,s}\}_{c,s}$ in all classes. Formally, *worst-group* test error is defined as:

$$\text{Err}_{wg} = \max_{g \in \mathcal{G}} \mathop{\mathbb{E}}_{(\boldsymbol{x}_i, y_i) \in g} [y_i \neq y_f(\boldsymbol{x}_i; \boldsymbol{W}, \boldsymbol{z})], \tag{4}$$

where $y_f(\boldsymbol{x}_i; \boldsymbol{W}, \boldsymbol{z})$ is the label predicted by the model. That is, $\text{Err}_{wg}$ measures the highest fraction of examples that are incorrectly classified across all groups.

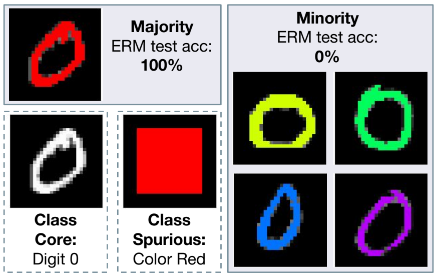


Figure 1: Colored MNIST as an example of datasets containing spurious correlations. Each digit is a class; the majority of digits in a class have a particular color, and the remaining digits are in other colors. Models trained with ERM learn to rely on spurious features (colors) instead of the core feature (digits) and thus do not perform well on groups of examples where the spurious correlation does not hold.

[1] Detailed assumptions on the activations, and initialization can be found in Sec. A

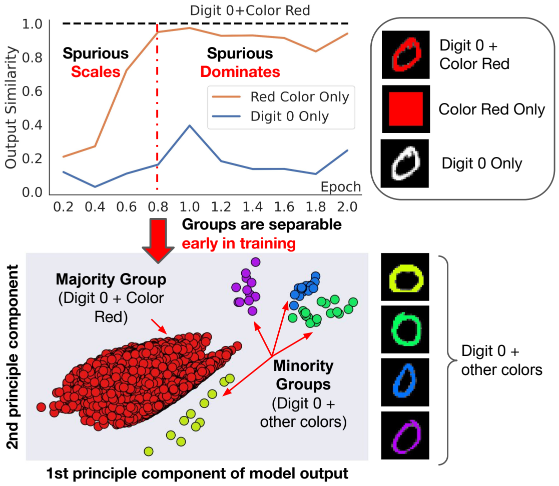


Figure 2: Training LeNet-5 on Colored MNIST. **Top**: Up to epoch 2, the network output is almost exclusively indicated by the color red (spurious feature in the majority group). **Bottom**: Majority and minority groups are separable based on the network output, e.g. via clustering. Minority groups that have a spurious feature in majority groups of other classes (yellow, purple, blue, green) are also separable from each other. Similar results on Waterbirds are shown in Fig. 6.

# 4 INVESTIGATING SPURIOUS FEATURE LEARNING IN NEURAL NETWORKS

We start by investigating how spurious features are learned during training a two-layer fully-connected neural network. Our analysis reveals two phases in early-time learning. First, in the initial training iterations, the contribution of a spurious feature to the network output increases linearly with the amount of the spurious correlation. Interestingly, if the majority group is sufficiently large, majority and minority groups are separable at this phase by the network output. Second, if the noise-to-signal ratio of the spurious feature of the majority group is smaller than that of the core feature, the network's output on the majority of examples in the class will be almost exclusively determined by the spurious feature and will remain mostly invariant to the core feature. Next, we will discuss the two phases in detail.

## 4.1 Spurious Features are Learned in the Initial Training Iterations

We start by analyzing the effect of spurious features on learning dynamics of a two-layer FC neural network trained with gradient descent in the initial training iterations. With the data model $\boldsymbol{x}_i = \boldsymbol{v}_c + \boldsymbol{v}_s + \boldsymbol{\xi}_i$ defined in Sec. 3, the following theorem shows that if a majority group is sufficiently large, contribution of its spurious feature to the model's output is magnified by the network at every step early in training.

**Theorem 4.1.** *Let $\alpha \in (0, \frac{1}{4})$ be a fixed constant. Suppose the number of training samples $n$ and the network width $m$ satisfy $n \gtrsim d^{1+\alpha}$ and $m \gtrsim d^{1+\alpha}$. Let $n_c$ be the number of examples in class $c$, and $n_{c,s} = |g_{c,s}|$ be the size of group $g_{c,s}$ with label $c$ and spurious feature $\boldsymbol{v}_s \in \mathcal{A}$. Then, under the setting of Sec. 3 there exist a constant $\nu_1 > 0$, such that with high probability, for all $0 \leq t \leq \nu_1 \cdot \sqrt{\frac{d^{1-\alpha}}{\eta}}$, the contribution of the core and spurious features to the network output can be quantified as follows:*

$$f(\boldsymbol{v}_c; \boldsymbol{W}_t, \boldsymbol{z}_t) = \frac{2\eta\zeta^2 c\|\boldsymbol{v}_c\|^2 t}{d}\left(\frac{n_c}{n} \pm O(d^{-\Omega(\alpha)})\right), \quad (5)$$

$$f(\boldsymbol{v}_s; \boldsymbol{W}_t, \boldsymbol{z}_t) = \frac{2\eta\zeta^2 c\|\boldsymbol{v}_s\|^2 t}{d}\Bigg(\frac{n_{c,s} - n_{c',s}}{n} \pm O(d^{-\Omega(\alpha)})\Bigg), \quad (6)$$

*where $c' = C \setminus c$, and $\zeta$ is the expected gradient of activation functions at random initialization.*

The proof for Theorem 4.1 is detailed in Sec. B.2. At a high level, as the model is nearly linear in the initial $\nu_1 \cdot \frac{d \log d}{\eta}$ iterations, the contribution of the spurious feature $\boldsymbol{v}_s$ to the network output grows almost linearly with $(n_{c,s} - n_{c',s})\|\boldsymbol{v}_s\|^2$, at *every iteration* in the initial phase of training. Here, $n_{c,s} - n_{c',s}$ represents the correlation between the spurious feature and label $c$. When $n_{c,s} \gg n_{c',s}$, the spurious feature exists almost exclusively in the majority group of class $c$, and thus has a high correlation only with class $c$. In this case, if the magnitude of the spurious feature is significant, the contribution of the spurious feature to the model's output grows very rapidly, early in training. In particular, if $(n_{c,s} - n_{c',s})\|\boldsymbol{v}_s\|^2 \gg n_c\|\boldsymbol{v}_c\|^2$, the model's output is increasingly determined by the spurious feature, but not the core feature.

Remember from Sec. 3 that every example consists of a core and a spurious feature. As the effect of spurious features of the majority groups is amplified in the network output, the model's output will differ for examples in the majority and minority groups. The following corollary shows that the majority and minority groups are separable based on the network's output early in training. Notably, multiple minority groups with spurious features contained in majority groups of other classes are also separable.

**Corollary 4.2** (**Separability of majority and minority groups**). *Suppose that for all classes, a major-*

*ity group has at least $K$ examples and a minority group has at most $k$ examples. Then, under the assumptions of Theorem 4.1, examples in the majority and minority groups are nearly separable with high probability based on the model's output, early in training. That is, for all $0 \leq t \leq \nu_1 \cdot \sqrt{\frac{d^{1-\alpha}}{\eta}}$, with high probability, the following holds for at least $1 - O(d^{-\Omega(\alpha)})$ fraction of the training examples $\boldsymbol{x}_i$ in group $g_{c,s}$:*

*If $g_{c,s}$ is in a majority group in class $c = 1$:*

$$f(\boldsymbol{x}_i; \boldsymbol{W}_t, \boldsymbol{z}_t) \geq \frac{2\eta\zeta^2 t}{d}\Bigg(\frac{\|\boldsymbol{v}_s\|^2(K-k)}{n} + \xi \pm O(d^{-\Omega(\alpha)})\Bigg) + \rho(t, \phi, \Sigma), \tag{7}$$

*If $g_{c,s}$ is in a minority group in class $c = 1$, but $g_{c',s}$ is a majority group in class $c' = -1$:*

$$f(\boldsymbol{x}_i; \boldsymbol{W}_t, \boldsymbol{z}_t) \leq \frac{2\eta\zeta^2 t}{d}\Bigg(-\frac{\|\boldsymbol{v}_s\|^2(K-k)}{n} + \xi \pm O(d^{-\Omega(\alpha)})\Bigg) + \rho(t, \phi, \Sigma), \tag{8}$$

*where $\rho$ is constant for all examples in the same class, $\xi \sim \mathcal{N}(0, \kappa)$ with $\kappa = \frac{1}{n}(\sum_c n_c^2\sigma_c^2\|\boldsymbol{v}_c\|^2)^{1/2} + \frac{1}{n}(\sum_s (n_{c,s} - n_{c',s})^2\sigma_s^2\|\boldsymbol{v}_s\|^2)^{1/2}$ is the total effect of noise on the model.*

*Analogous statements holds for the class $c = -1$ by changing the sign and direction of the inequality.*

The proof can be found in Sec. B.2. Corollary 4.2 shows that when the majority group is considerably larger than the minority groups ($K \gg k$), the prediction of examples in the majority group move toward their label considerably faster, due to the contribution of the spurious feature. Hence, majority and minority groups can be separated from each other, early in training. Importantly, multiple minority groups can be also separated from each other, if their spurious feature exists in majority groups of other classes. Note that $K > k + |\xi|$ is the minimum requirement for the separation to happen. Separation is more significant when $K \gg k$ and when $\|\boldsymbol{v}_s\|$ is significant.

## 4.2 Network Relies on Simple Spurious Features for Majority of Examples

Next, we analyze the second phase in early-time learning of a two-layer neural network. In particular, we show that if the noise-to-signal ratio of the spurious feature of the majority group of class $c$, i.e., $R_s = \sigma_s/\|\boldsymbol{v}_s\|$ is smaller than that of the core feature $R_c = \sigma_c/\|\boldsymbol{v}_c\|$, then the neural network's output is almost exclusively determined by the spurious feature and remain invariant to the core feature at $T = \nu_2 \cdot \frac{d \log d}{\eta}$, even though the core feature is more predictive of the class.

**Theorem 4.3.** *Under the assumptions of Theorem 4.1, if the classes are balanced, and the total size of the minority groups in class $c$ is small, i.e., $O(n^{1-\gamma})$ for some $\gamma > 0$, then there exists a constant $\nu_2 > 0$ such that at $T = \nu_2 \cdot \frac{d \log d}{\eta}$, for an example $\boldsymbol{x}_i$ in a majority group $g_{c,s}$, the contribution of the core feature to the model's output is at most:*

$$|f(\boldsymbol{v}_c; \boldsymbol{W}_T, \boldsymbol{z}_T)| \leq \Bigg(\frac{\sqrt{2}R_s}{R_c} + O(n^{-\gamma} + d^{-\Omega(\alpha)})\Bigg). \tag{9}$$

*In particular if $\min\{R_c, 1\} \gg R_s$, then the model's output is mostly indicated by the spurious feature instead of the core feature:*

$$|f(\boldsymbol{v}_s; \boldsymbol{W}_T, \boldsymbol{z}_T)| \gg |f(\boldsymbol{v}_c; \boldsymbol{W}_T, \boldsymbol{z}_T)|. \tag{10}$$

The proof can be found in Sec. B.3. Theorem 4.3 shows that at $T = \nu_2 \cdot \frac{d \log d}{\eta}$, the contribution of the core feature to the network's output is at most proportional to $R_s/R_c$. Hence, if $R_s \ll R_c$, the network almost exclusively relies on the spurious feature of the majority group instead of the core feature.

We note that our results in Theorem 4.1, Corollary 4.2, and Theorem 4.3 generalize to more than two classes and hold if the classes are imbalanced, as we will confirm by our experiments. Similar results can be shown for multi-layer fully connected and convolutional networks, following (Hu et al., 2020).

## 4.3 Separability of Majority and Minority Groups in The Two Early Training Phases

The initial phase of training iterations, when $0 \leq t \leq \nu_1 \cdot \sqrt{\frac{d^{1-\alpha}}{\eta}}$, is characterized by approximately linear change in the loss. Corollary 4.2 shows that in Phase 1, majority and minority groups are separable based on the network output, if spurious feature is strongly correlated with label and has a higher magnitude than the core. Phase 2 happens when $T = \nu_2 \cdot \frac{d \log d}{\eta}$ and marks the point where the approximate linear model converges to its optimal parameters, and the network starts learning higher-order (non-linear) functions. Theorem 4.3 shows that in Phase 2, majority and minority groups are separable based on the network output, if the noise-to-signal ratio of spurious feature ($R_s$) is smaller than core ($R_c$).

The above discussion implies that one can separate majority and minority groups in Phase 1 or Phase 2, as long as the corresponding conditions are met.

**Visualization of Theoretical Results.** We empirically illustrate the above results during early-time

**Algorithm 1** SePArate early and REsample (Spare)

```
Input: Network $f(.,\boldsymbol{W})$, data $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, loss
  function $\mathcal{L}$, iteration numbers $T_N, T_{init}$.
Output: Model $f$ trained without bias
  Stage 1: Early Bias Identification
  for $t = 0, \cdots, T_{init}$ do
    $\boldsymbol{W}_{t+1} \leftarrow \boldsymbol{W}_t - \eta \nabla \mathcal{L}(\boldsymbol{W}_t; \mathcal{D})$
  end for
  for every class $c \in \mathcal{C}$ with examples $V_c$ do
    Identify $\lambda$, # of clusters $k$ via Silhouette analysis
    Cluster $V_c$ into $\{V_{c,j}\}_{j=1}^k$ based on $f(\boldsymbol{x}_i; \boldsymbol{W}_t)$
    Weight every $\boldsymbol{x}_i \in V_{c,j}$ by $w_i = 1/|V_{c,j}|$,
    $p_i = w_i^\lambda / \sum_i w_i^\lambda$
  end for
  Stage 2: Learning without Bias
  for $t = 0, \cdots, T_N$ do
    Sample a mini-batch $\mathcal{M}_t = \{(\boldsymbol{x}_i, y_i)\}_i$ with
    probabilities $p_i$
    $\boldsymbol{W}_{t+1} = \boldsymbol{W}_t - \eta \nabla \mathcal{L}(\boldsymbol{W}_t; \mathcal{M}_t)$.
  end for
```

training of LeNet-5 (LeCun et al., 1998) on the Colored MNIST (Alain et al., 2015) dataset containing colored handwritten digits. Fig. 2 shows that the prediction of the network on the majority group is almost exclusively indicated by the color of the majority group, confirming Theorem 4.3. The bottom of Fig. 2 shows that the majority and minority groups are separable based on the network output, confirming Corollary 4.2.

**Strong Spurious Correlations Make the Network Invariant to Core Features of Majority Groups.** Finally, note that by only learning the spurious feature, the neural network can shrink the training loss on the majority of examples in class $c$ to nearly zero and correctly classify them. Hence, the contribution of the spurious feature of the majority group of class $c$ to the model's output is retained throughout the training. On the other hand, if minority groups are small, higher complexity functions that appear later in training overfit the minority groups, as observed by (Sagawa et al., 2020). This results in a small training error but a poor worst-group generalization performance on the minorities.

## 5 SPARE: ELIMINATING SPURIOUS BIAS EARLY IN TRAINING

Based on the theoretical foundations outlined in Sec. 4, we develop a principled pipeline, Spare, to discover and mitigate spurious correlations *early in training.* The pseudocode is illustrated in Alg. 1.

**Discovering Spurious Correlations: Separating the Groups Early in Training.** Corollary 4.2 shows that majority and minority groups are separable based on the network's output. To identify the majority and minority groups, we cluster examples $V_c$ in every class $c \in \mathcal{C}$ based on the output of the network, during the first few epochs. We tune $T_{init}$ for maximum recall of Spare's clusters against the validation set groups in the first 1-2 epochs (discussed in Sec. E.2). We determine the number of clusters via silhouette analysis (Rousseeuw, 1987), a technique that assesses the cohesion and separation of clusters by evaluating how close each point in one cluster is to points in the neighboring clusters. In doing so, we can separate majority and minority groups in each class of examples with different spurious features. Any clustering algorithm such as $k$-means or $k$-median clustering (Mirzasoleiman et al., 2013, 2015) can be applied to separate groups in large data.

**Mitigation after Discovery: Balancing Groups via Importance Sampling.** To alleviate the spurious correlations and enable effective learning of the core features, we employ an importance sampling method on examples in each class to upsample examples in the smaller clusters and downsample examples in the larger clusters. To do so, we assign every example $i \in V_{c,j}$ a weight given by the size of the cluster it belongs to, i.e., $w_i = 1/|V_{c,j}|$. Then we sample examples in every mini-batch with probabilities equal to $p_i = w_i^\lambda / \sum_i w_i^\lambda$, where $\lambda$ can be determined based on the average silhouette score of clusters in each class, *without further tuning.* Our importance sampling method does not increase the size of the training data but only changes the data distribution. Hence, it does not increase the training time.

## 6 EXPERIMENTS

In this section, we first confirm that Spare outperforms state-of-the-art baselines in identifying and mitigating spurious correlations across multiple curated benchmark datasets. Most notably, Spare excels on UrbanCars, a challenging dataset with multiple spurious correlations within each class. Then, we demonstrate that Spare effectively discovers and mitigates naturally occurring spurious correlations early in training on Restricted ImageNet—a realistic dataset *without known spurious correlations.*

### 6.1 Mitigating Curated Spurious Correlations in Benchmark Datasets

First, we evaluate the effectiveness of Spare in alleviating spurious correlations on spurious benchmarks. The reported results are averaged over three runs with different model initializations.

Table 1: Worst-group and average accuracy (%) of training with Spare vs. state-of-the-art algorithms. Spare achieves a superior performance much faster that existing methods. CB, GB indicate balancing classes and groups. Range for training cost encompasses all datasets, and accounts for (1) training the reference model for group inference and (2) number of training examples involved in robust training (excluding tuning cost). Baseline results for CMNIST, UrbanCars are from the benchmarks (Zhang et al., 2022; Li et al., 2023). ♦ and △ indicate using group-labeled validation for tuning group inference, and robust training. (E#) shows the early group inference epoch for Spare. (◊) shows Spare doesn't heavily rely on validation set (◊). We couldn't successfully run CnC on UrbanCars and DFR on CMNIST.

| | Grp label required | Train cost | **CMNIST**<br>(1 Spurious × **5 Classes**) | | **Waterbirds**<br>(1 Spurious × 2 Classes) | | **CelebA**<br>(1 Spurious × 2 Classes) | | **UrbanCars**<br>(**2 Spurious** × 2 Classes) | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Worst-group | Average | Worst-group | Average | Worst-group | Average | Worst-group | Average |
| ERM | −− | 1x | $0.0_{\pm0.0}$ | $20.1_{\pm0.2}$ | $62.6_{\pm0.3}$ | $97.3_{\pm1.0}$ | $47.7_{\pm2.1}$ | $94.9_{\pm0.3}$ | 28.4 | 97.6 |
| CB | −− | 1x | $0.0_{\pm0.0}$ | $23.7_{\pm3.1}$ | $62.8_{\pm1.6}$ | $97.1_{\pm0.1}$ | $46.1_{\pm1.5}$ | $95.2_{\pm0.4}$ | 33.7 | 98.1 |
| George (Sohoni et al., 2020) | −− | 2x | $76.4_{\pm2.3}$ | $89.5_{\pm0.3}$ | $76.2_{\pm2.0}$ | $95.7_{\pm0.5}$ | $54.9_{\pm1.9}$ | $94.6_{\pm0.2}$ | 35.2 | 97.9 |
| PGI (Ahmed et al., 2020) | ♦− | 1x | $73.5_{\pm1.8}$ | $88.5_{\pm1.4}$ | $79.5_{\pm1.9}$ | $95.5_{\pm0.8}$ | $85.3_{\pm0.3}$ | $87.3_{\pm0.1}$ | 34.0 | 95.7 |
| EIIL (Creager et al., 2021) | ♦− | 1x | $72.8_{\pm\mathbf{6.8}}$ | $90.7_{\pm0.9}$ | $83.5_{\pm2.8}$ | $94.2_{\pm1.3}$ | $81.7_{\pm0.8}$ | $85.7_{\pm0.1}$ | 50.6 | 95.5 |
| LfF (Nam et al., 2020) | ♦△ | 2x | $0.0_{\pm0.0}$ | $25.0_{\pm0.5}$ | $78.0_{N/A}$ | $91.2_{N/A}$ | $77.2_{N/A}$ | $85.1_{N/A}$ | 34.0 | 97.2 |
| JTT (Liu et al., 2021) | ♦△ | 5x-6x | $74.5_{\pm\mathbf{2.4}}$ | $90.2_{\pm0.8}$ | $83.1_{\pm\mathbf{3.5}}$ | $90.6_{\pm0.3}$ | $81.5_{\pm1.7}$ | $88.1_{\pm0.3}$ | 55.8 | 95.9 |
| CnC (Zhang et al., 2022) | ♦△ | 2x-12x | $77.4_{\pm3.0}$ | $90.9_{\pm0.6}$ | $88.5_{\pm0.3}$ | $90.9_{\pm0.1}$ | $88.8_{\pm0.9}$ | $89.9_{\pm0.5}$ | - | - |
| **Spare**<br>(−$2^{nd}$ best) | ◊− | **1x** | **(E2)** $\mathbf{83.0}_{\pm1.7}$<br>**(+5.6)** | $91.8_{\pm0.7}$ | **(E2)** $\mathbf{91.6}_{\pm0.8}$<br>**(+3.1)** | $96.2_{\pm0.6}$ | **(E1)** $\mathbf{90.3}_{\pm0.3}$<br>**(+1.5)** | $91.1_{\pm0.1}$ | **(E2)** $\mathbf{76.9}_{\pm1.8}$<br>**(+21.1)** | $96.6_{\pm0.5}$ |
| DFR (Kirichenko et al., 2023) | train sub | 1x | - | - | $90.4_{\pm1.5}$ | $94.1_{\pm0.5}$ | $80.1_{\pm1.1}$ | $89.7_{\pm0.4}$ | 44.5 | 89.7 |
| GB | train full | 1x | $82.2_{\pm1.0}$ | $91.7_{\pm0.6}$ | $86.3_{\pm0.3}$ | $93.0_{\pm1.5}$ | $85.0_{\pm1.1}$ | $92.7_{\pm0.1}$ | 73.9 | 92.2 |
| GDRO (Sagawa et al., 2019) | train full | 1x | $78.5_{\pm4.5}$ | $90.6_{\pm0.1}$ | $89.9_{\pm0.6}$ | $92.0_{\pm0.6}$ | $88.9_{\pm1.3}$ | $93.9_{\pm0.1}$ | 75.2 | 91.6 |

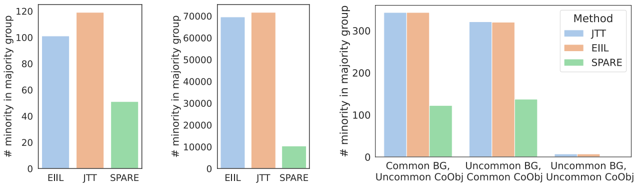
**Benchmark Datasets & Models.** (1) CMNIST (Alain et al., 2015) contains colored handwritten digits derived from MNIST (LeCun et al., 1998). We follow the challenging 5-class setting in Zhang et al. (2022) where every two digits form one class and 99.5% of training examples in each class are spuriously correlated with a distinct color. We use a 5-layer CNN (LeNet-5 (LeCun et al., 1998)) for CMNIST. (2) Waterbirds (Sagawa et al., 2019) contains two classes (landbird vs. waterbird) and the background (land or water) is the spurious feature. Majority groups are (waterbird, water) and (landbird, land). (3) CelebA (Liu et al., 2015) is a face datasets containing two classes (blond vs dark) and gender (male or female) as the spurious feature (Sagawa et al., 2019). Majority groups are (blond, female) and (non-blond, male). (4) UrbanCars (Li et al., 2023) is a challenging task containing two classes (urban vs. country cars) and *two spurious features*: (1) background (BG): urban vs. country and (2) co-occurring object (CoObj): fireplug and stop sign vs cow and horse. For Waterbirds, CelebA, and UrbanCars, we follow the standard settings in previous work to train a ResNet-50 model (He et al., 2016) pretrained on ImageNet provided by the Pytorch library (Paszke et al., 2019). More details about the datasets and the experimental settings are in Sec. E.

**Baselines.** We compare Spare with the state-of-the-art methods for eliminating spurious correlations in Tab. 1, in terms of both worst-group and average accuracy. We use adjusted average accuracy for Waterbirds, i.e., the average accuracy over groups weighted by their size. This is consistent with prior work, and is done because the validation and test sets are group-balanced while the training set is skewed. For UrbanCars, our average accuracy corresponds to the ID accuracy in the original paper (Li et al., 2023), and our worst-group accuracy is computed by adding the largest of BG/CoObj/BG+CoObj gap to the ID accuracy. GB (Group Balancing) and GDRO (Sagawa et al., 2019) use the group label of all training examples. DFR (Kirichenko et al., 2023) uses group-balanced data drawn from either validation ($\text{DFR}^{Tr}_{Tr}$) or training ($\text{DFR}^{Val}_{Tr}$) data. We considered $\text{DFR}^{tr}_{tr}$, which trains on group-balanced training data, for a fair comparison with baselines that only use the validation set to tune. The rest of the methods infer the group labels without using such information.

**Spare Outperforms SOTA Algorithms, Especially When Multiple Spurious Correlations Exist.** Tab. 1 shows that Spare consistently outperforms the best baselines, on worst-group and average accuracy. On UrbanCars, where baselines have been shown to amplify one spurious when trying to mitigate the other (Li et al., 2023), Spare achieves a 21.1% higher accuracy than the next best state-of-the-art method that does not rely on ground-truth group labels during training. Notably, Spare performs comparably to those that *use* the group information, and even achieves a better worst-group accuracy on CMNIST, CelebA, and UrbanCars, and has a comparable worst-group but higher average accuracy on the Waterbirds. As group labels are unavailable in real-world datasets, methods that do not rely on group labels are more practical. Moreover, group labels can sometimes

| Method | BG (↑) | CoObj (↑) | BG+CoObj (↑) |
|---|---|---|---|
| JTT (E1) | -8.1 | -13.3 | -40.1 |
| EIIL (E1) | -4.2 | -24.7 | -44.9 |
| JTT (E2) | -23.3 | -5.3 | -52.1 |
| EIIL (E2) | -21.5 | -6.8 | -49.6 |
| **Spare** (E2) | **-5.3** | **-3.1** | **-8.9** |

Table 2: UrbanCars with two spurious correlations (BG and CoObj). SOTA methods show "whack-a-mole" behavior (Li et al., 2023): mitigating one spurious correlation amplifies the other, Spare finds minority groups more accurately and does not exhibit whack-a-mole.



(a) Waterbirds (b) CelebA (c) UrbanCars

Figure 3: Number of minority examples inferred as majority. JTT and EIIL infer many minority examples as majority and mistakenly downweight them. Spare identifies minority groups more accurately, and correctly upweights them.

be inaccurate, so group-inference methods like Spare can better identify the groups. This aligns with EIIL's observation that inferred groups can be more informative than underlying group labels (Creager et al., 2021). Among such methods, Spare has a superior performance and easily scales to large datasets.

**Spare Does Not Exhibit Whack-a-mole Behavior when Data Contains Multiple Spurious Correlations.** Tab. 2 shows the accuracy of different groups in UrbanCars datasets with two spurious correlations (BG and CoObj). We see that JTT and EIIL drop the accuracy on BG or CoObj when trying to improve the accuracy on the other. This is because they find a smaller minority group, and hence need to upweight it heavily to mitigate the spurious correlation. In doing so, they amplify the other subtle spurious correlation in the minority group. In contrast, Spare finds minority groups more accurately and mitigates the spurious correlation without introducing a new one. Notably, Spare achieves 31.2% better accuracy than the best baseline on the smallest group (BG+CoObj).

**Spare is Much Faster and Easier to Tune.** As our theoretical results narrow the range for inference time to the initial epochs (often 1 or 2 in practice), it can be tuned easily for Spare, while others need to search over a wide range from epoch 1 to 60. In addition to the time saved for hyperparameter tuning, Spare has up to 12x lower computational cost ($k$-means & total wall-clock runtimes are reported in Tab. 6 and Tab. 7) compared to the state-of-the-art.

## 6.2 Why Does Spare Work Better?

**Spare Finds Minority Groups More Accurately.** Fig. 3 explains the superior performance of Spare over the state-of-the-art: JTT (Liu et al., 2021) and EIIL (Creager et al., 2021) mistakenly group many minority examples into majority groups and thus mistakenly downweight them. The minority samples include, by definition, all instances where spurious correlations are not present. In contrast, by finding the minority groups more accurately, Spare effectively upweights them to mitigate the spurious correlations and improve the worst-group accuracy. We also compare the worst-group and average accuracy of models trained with GDRO and JTT's upsampling method applied to groups inferred by Spare in Tab. 3. Comparing to Tab. 1, we see that both methods obtain a better worst-group accuracy using Spare's groups. In particular, training on Spare's groups with GDRO outperform George and EIIL that use GDRO by 11.4%, 4.1%. Similarly, training on Spare's groups with JTT's upsampling outperforms JTT by 4.5%, further confirming that Spare finds minorities more accurately.

Table 3: Spare's importance sampling is more effective in improving the worst-group accuracy than GDRO and JTT's upsampling, when applied to Spare's groups.

| Groups | Robust training | Worst-group | Avg Acc |
|---|---|---|---|
| Spare | JTT | 86.2 ± 3.6 | 92.0 ± 0.8 |
| Spare | GDRO(/George/EIIL) | 87.6 ± 0.8 | 89.4 ± 1.3 |
| Spare | Spare | **91.6 ± 0.8** | **96.2 ± 0.6** |

**Spare's Importance Sampling is More Effective and Efficient.** Next, we compare the worst-group and average accuracy of models trained with GDRO, JTT's upsampling, and Spare's importance sampling, applied to groups inferred by Spare. Tab. 3 shows that Spare's importance sampling is more effective in improving the worst-group accuracy and outperforms GDRO by 4% and JTT's upsampling by 5.4%. Note that both methods require tuning based on group-labeled validation data, and upsampling drastically increases the training time. On the other hand, Spare's importance sampling does not require any hyperparameter tuning or increase the training time.

## 6.3 Discovering Natural Spurious Correlations in Restricted ImageNet

Next, we show the applicability of Spare to discover and mitigate spurious correlations in Restricted Ima-

Table 4: Discovering & mitigating spurious correlations in Restricted ImageNet. SPARE infers groups more accurately and improves both insect and frog minority accuracy by 1.2% and 11.5% respectively, with only a minor drop in total accuracy. Note that the group with worst-group accuracy changes during the training.

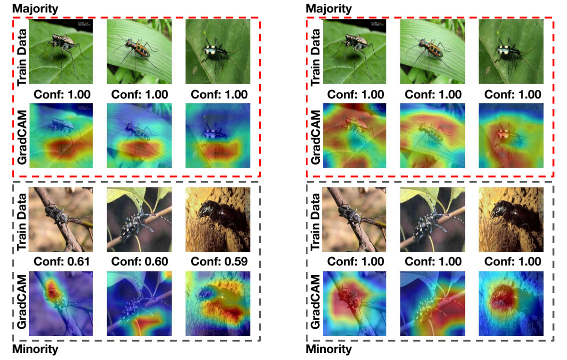
| Method | Minority Recall | Test Avg Acc | Insect Min Acc | Frog Min Acc | W-G Acc |
|---|---|---|---|---|---|
| ERM | - | 96.0% | 91.7% | 80.8% | 80.8% |
| CB | - | 95.9% | **93.7%** ↑ | 80.8%– | 80.8%– |
| EIIL | 78.8% | 93.1% | 88.3% ↓ | 69.2% ↓ | 69.2% ↓ |
| JTT | 82.6% | 92.8% | 75.0% ↓ | **92.3%** ↑ | 75.0% ↓ |
| GEORGE | 85.4% | 94.8% | 89.4% ↓ | 80.8%– | 80.8%– |
| **SPARE** | **92.8%** | 95.4% | **92.9%** ↑ | **92.3%** ↑ | **92.3%** ↑ |

geNet (Taghanaki et al., 2021), a 9-superclass subset of ImageNet. Here, we train ResNet-50 from scratch. See G for more details on the dataset and experiment.

**SPARE Discovers Spurious Correlations** We observe SPARE clusters during the initial training epochs. Inspecting the clusters with the highest fraction of misclassified examples to another class, we find that many frog images are misclassified as insects. Fig. 4a shows examples from the two groups SPARE finds for the Insect class at epoch 8, where clusters with spurious features are visually evident [2]. GradCAM reveals an obvious spurious correlation between "green leaf" and the insect class that is maintained until the end of the training, as illustrated in Fig. 4b. We also observe a large gap between the confidence of examples in the two groups. This indicates that the model has learned the spurious feature early in training.

**SPARE Discovers Spurious Correlations without Relying on Group-labeled Validation Data.** State-of-the-art group inference baselines heavily rely on a group-labeled validation set to identify the time of group inference during training with ERM. While SPARE can also benefit from a group-labeled validation, this is not essential. In fact, our theoretical results reveal that the range for inference time should be within the initial epochs. Thus we only visually inspect a few initial epochs (2, 4, 6, 8) to verify the spurious correlations. This sets SPARE apart as a more generally applicable method for discovering and mitigating spurious correlations, even in the absence of a group-labeled validation set.

**SPARE Achieves State-of-the-art Accuracy on Minority Groups.** Based on the spurious correlations SPARE discovered, we manually labeled the background of both training and test data for the *insect* and *frog* classes, and these group labels to tune the baseline group inference methods. Tab. 4 shows SPARE separates the insect majority group with the spurious correlation better than other group inference methods and improves both insect and frog minority accuracy by 1.2% and 11.5% respectively, with only a minor drop in total accuracy. Note that group with worst-group accuracy changes during the training. CB only improves insect minority accuracy. JTT decreases the model's accuracy on the insect minority a lot while improving the frog minority. EIIL decreases both minority and total accuracy as it finds the least majority. Unlike the baselines, SPARE effectively balances groups, mitigating spurious correlations.



(a) ERM epoch 8: model learns the spurious feature "green leaf" in Insect class.

(b) ERM end of training: model keep relying on "green leaf", instead of Insect.

Figure 4: SPARE-discovered spurious correlation between "green leaf" & "insect" in Restricted ImageNet.

## 7 CONCLUSION

We studied the effect of simplicity bias of SGD on learning spurious features. Specifically, we analyzed a two-layer fully-connected neural network and showed that spurious features can be identified early in training based on model output. If spurious features have a low noise-to-signal ratio, they dominate the network's output, overshadowing core features. Based on the above theoretical insights, we proposed SPARE, which separates majority and minority groups by clustering the model output early in training. Then, it applies importance sampling based on the cluster sizes to make the groups relatively balanced. It outperforms state-of-the-art methods in worst-group accuracy on benchmark datasets and scales well to large-scale applications. Importantly, it can deal with multiple spurious correlations, minimizes the need for hyperparameter tuning, and can discover spurious correlations in realistic scenarios like Restricted ImageNet, early in training.

[2] Since the model is not pretrained, it is expected that the spurious clusters form slightly later. For pretrained models, spurious clusters form very early, as shown in Tab. 1

## Acknowledgements

This research was partially supported by the National Science Foundation CAREER Award 2146492, Amazon PhD fellowship, and Google Cloud research credits.

## References

F. Ahmed, Y. Bengio, H. van Seijen, and A. Courville. Systematic generalisation with group invariant predictions. In *International Conference on Learning Representations*, 2020.

G. Alain, A. Lamb, C. Sankar, A. Courville, and Y. Bengio. Variance reduction in sgd by distributed importance sampling. *arXiv preprint arXiv:1511.06481*, 2015.

M. Arjovsky and L. Bottou. Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862*, 2017.

A. Badanidiyuru, B. Mirzasoleiman, A. Karbasi, and A. Krause. Streaming submodular maximization: Massive data summarization on the fly. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 671–680, 2014.

H. Cramér. *Mathematical methods of statistics*, volume 43. Princeton university press, 1999.

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

P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

A. Gupta, P. Dollar, and R. Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

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

K. Hermann and A. Lampinen. What shapes feature representations? exploring datasets, architectures, and training. *Advances in Neural Information Processing Systems*, 33:9995–10006, 2020.

W. Hu, L. Xiao, B. Adlam, and J. Pennington. The surprising simplicity of the early-time learning dynamics of neural networks. *Advances in Neural Information Processing Systems*, 33:17116–17128, 2020.

P. Kirichenko, P. Izmailov, and A. G. Wilson. Last layer re-training is sufficient for robustness to spurious correlations. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=Zb6c8A-Fghk`.

J. Krause, M. Stark, J. Deng, and L. Fei-Fei. 3d object representations for fine-grained categorization. In *2013 IEEE International Conference on Computer Vision Workshops*, pages 554–561, 2013. doi: 10.1109/ICCVW.2013.77.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Z. Li, I. Evtimov, A. Gordo, C. Hazirbas, T. Hassner, C. C. Ferrer, C. Xu, and M. Ibrahim. A whac-a-mole dilemma: Shortcuts come in multiples where mitigating one amplifies others. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20071–20082, 2023.

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

Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

B. Mirzasoleiman, A. Karbasi, R. Sarkar, and A. Krause. Distributed submodular maximization: Identifying representative elements in massive data. In *Advances in Neural Information Processing Systems*, pages 2049–2057, 2013.

B. Mirzasoleiman, A. Badanidiyuru, A. Karbasi, J. Vondrák, and A. Krause. Lazier than lazy greedy. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

V. Nagarajan, A. Andreassen, and B. Neyshabur. Understanding the failure modes of out-of-distribution generalization. *arXiv preprint arXiv:2010.15775*, 2020.

P. Nakkiran, G. Kaplun, D. Kalimeris, T. Yang, B. L. Edelman, F. Zhang, and B. Barak. Sgd on neural networks learns functions of increasing complexity. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 3496–3506, 2019.

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

J. Nam, J. Kim, J. Lee, and J. Shin. Spread spurious attribute: Improving worst-group accuracy with spurious attribute estimation. In *International Conference on Learning Representations*, 2021.

B. Neyshabur, R. Tomioka, and N. Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv preprint arXiv:1412.6614*, 2014.

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

M. Pezeshki, O. Kaba, Y. Bengio, A. C. Courville, D. Precup, and G. Lajoie. Gradient starvation: A learning proclivity in neural networks. *Advances in Neural Information Processing Systems*, 34:1256–1272, 2021.

P. J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20: 53–65, 1987.

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

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

M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

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

H. Shah, K. Tamuly, A. Raghunathan, P. Jain, and P. Netrapalli. The pitfalls of simplicity bias in neural networks. *Advances in Neural Information Processing Systems*, 33:9573–9585, 2020.

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

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

D. Teney, E. Abbasnejad, S. Lucey, and A. Van den Hengel. Evading the simplicity bias: Training a diverse set of models discovers solutions with superior ood generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16761–16772, 2022.

D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=SyxAb30cY7`.

C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

L. A. Wolsey. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 2(4):385–393, 1982.

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

B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, 2018. doi: 10.1109/TPAMI.2017.2723009.

## Checklist

1. For all models and algorithms presented, check if you include:
    (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
    (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
    (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
    (a) Statements of the full set of assumptions of all theoretical results. [Yes]
    (b) Complete proofs of all theoretical results. [Yes]
    (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
    (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
    (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
    (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
    (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
    (a) Citations of the creator If your work uses existing assets. [Yes]
    (b) The license information of the assets, if applicable. [Yes]
    (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
    (d) Information about consent from data providers/curators. [Not Applicable]
    (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
    (a) The full text of instructions given to participants and screenshots. [Not Applicable]
    (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
    (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

# A SIMPLICITY BIAS

A recent body of work revealed that the neural network trained with (stochastic) gradient methods can be approximated on the training data by a linear function early in training (Hermann and Lampinen, 2020; Hu et al., 2020; Nakkiran et al., 2019; Neyshabur et al., 2014; Pezeshki et al., 2021; Shah et al., 2020). We hypothesize that a slightly stronger statement holds, namely the approximation still holds if we isolate a core or spurious feature from an example and input it to the model.

**Assumption A.1** (simplicity bias on core and spurious features, informal)**.** Suppose that $f^{lin}$ is a linear function that closely approximates $f(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{z})$ on the training data. Then $f^{lin}$ also approximates $f$ on input either a core feature or a spurious feature corresponding to a majority group in some class, that is

$$f^{lin}(\boldsymbol{v}_c) \approx f(\boldsymbol{v}_c; \boldsymbol{W}, \boldsymbol{z}) \qquad \forall c \in C$$
$$f^{lin}(\boldsymbol{v}_s) \approx f(\boldsymbol{v}_s; \boldsymbol{W}, \boldsymbol{z}) \qquad \forall s \in \mathcal{A}$$

Intuitively, every core feature and every spurious feature corresponding to a majority group is well represented in the training dataset, and since it is known that the linear model and the full neural network agree on the training dataset, we can expect them to agree on such features as well. Note that spurious features that do not appear in majority groups may not be well represented in the training dataset, hence we do not require that the linear model approximates the neural network well on such features.

Moreover, we verify Assumption A.1 empirically on CMNIST in Fig. 5, which shows that a two-layer neural network and the approximating linear model are close even when isolating a core or spurious feature.

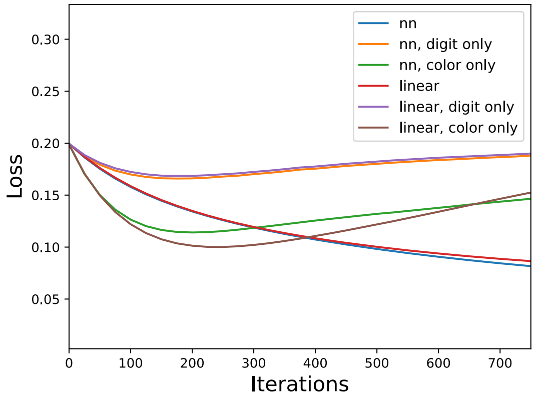


Figure 5: A comparison between the losses of a two-layer network and a simple linear model on the training set, spurious features (color only), and core feature (digit only).

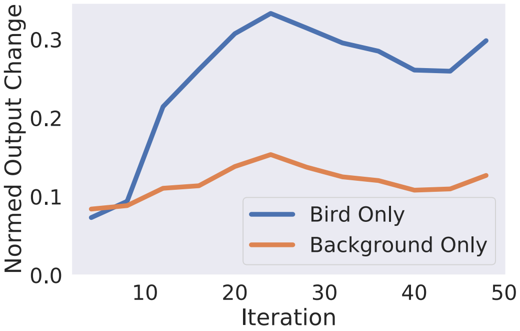


Figure 6: Replicate of Figure 1b on Waterbirds. Inputting only the background (orange line) does not change the model output much (indicating that the background is learned by the model) while inputting only the bird changes the output to a large extent (indicating that the bird is not learned by the model).

The formal statement is provided below as Assumption A.6.

**Setting** We now introduce the formal mathematical setting for the theory. Let $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n \subset \mathbb{R}^d \times \mathbb{R}$, be a dataset with covariance $\boldsymbol{\Sigma}$. Define the data matrix $\boldsymbol{X} = \begin{bmatrix}\boldsymbol{x}_1 & \dots & \boldsymbol{x}_n\end{bmatrix}^\top$ and the label vector $\boldsymbol{y} = \begin{bmatrix}y_1 & \dots & y_n\end{bmatrix}^\top$. We use $\|\cdot\|$ to refer to the Euclidean norm of a vector or the spectral norm of the data.

Following Hu et al. (2020), we make the following assumptions:

**Assumption A.2** (input distribution)**.** The data has the following properties (with high probability):

$$\frac{\|\boldsymbol{x}_i\|^2}{d} = 1 \pm O(\sqrt{\frac{\log n}{d}}), \forall i \in [n]$$
$$\frac{|\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle|}{d} = O(\sqrt{\frac{\log n}{d}}), \forall i, j \in [n], i \neq j$$
$$\|\boldsymbol{X}\boldsymbol{X}^\top\| = \Theta(n)$$

**Assumption A.3** (activation function)**.** The activation $\phi(\cdot)$ satisfies either of the following:

- smooth activation: $\phi$ has bounded first and second derivative
- piecewise linear activation:

$$\phi(z) = \begin{cases} z & z \geq 0 \\ az & z < 0 \end{cases}$$

**Assumption A.4** (initialization)**.** The weights $(\boldsymbol{W}, \boldsymbol{v})$ are initialized using symmetric initialization:

$$\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{\frac{m}{2}} \sim \mathcal{N}(\boldsymbol{0}_d, \boldsymbol{I}_d), \qquad \boldsymbol{w}_{i+\frac{m}{2}} = \boldsymbol{w}_i (\forall i \in 1, \ldots, \frac{m}{2})$$
$$v_1, \ldots, v_{\frac{m}{2}} \sim \text{Unif}(\{-1, 1\}), \qquad v_{i+\frac{m}{2}} = -v_i (\forall i \in 1, \ldots, \frac{m}{2})$$

It is not hard to check that the concrete scenario we choose in our analysis satisfies the above assumptions. Now, given the following assumptions, we leverage the result of (Hu et al., 2020):

**Theorem A.5** ((Hu et al., 2020))**.** *Let $\alpha \in (0, 1/4)$ be a fixed constant. Suppose $d$ is the input dimensionality, $\frac{\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle}{d} = \mathbb{1}_{i=j} \pm O(\sqrt{\frac{\log n}{d}}), \forall i, j \in [n]$, the data matrix $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^n$ has spectral norm $\|\boldsymbol{X}\boldsymbol{X}^\top\| = \Theta(n)$, and for the labels we have $|y_i| \leq 1\ \forall y_i$. Assume the number of training samples $n$ and the network width $m$ satisfy $n, m = \Omega(d^{1+\alpha}), n, m \leq d^{O(1)}$, and the learning rate $\eta \ll d$. Then, there exist a universal constant $C$, such that with high probability for all $0 \leq t \leq T = C \cdot \frac{d \log d}{\eta}$, the network $f(\boldsymbol{w}_t, \boldsymbol{X})$ trained with GD is very close to a linear function $f^{lin}(\boldsymbol{\beta}, \boldsymbol{X})$:*

$$\frac{1}{n} \sum_{i=1}^{n} (f^{lin}(\boldsymbol{\beta}_t, \boldsymbol{X}) - f(\boldsymbol{w}_t, \boldsymbol{X}))^2 \leq \frac{\eta^2 t^2}{d^{2+\Omega(\alpha)}} \leq \frac{1}{d^{\Omega(\alpha)}}. \tag{11}$$

In particular, the linear model $f^{lin}(\boldsymbol{\beta}, \boldsymbol{X})$ operates on the transformed data $\boldsymbol{\psi}(\boldsymbol{x})$, where

$$\boldsymbol{\psi}(\boldsymbol{x}) = \begin{bmatrix} \sqrt{\frac{2}{d}} \zeta \boldsymbol{x} \\ \sqrt{\frac{3}{2d}} \nu \\ \vartheta_0 + \vartheta_1 (\frac{\|\boldsymbol{x}\|}{\sqrt{d}} - 1) + \vartheta_2 (\frac{\|\boldsymbol{x}\|}{\sqrt{d}} - 1)^2 \end{bmatrix}$$
$$\zeta = \mathop{\mathbb{E}}_{g \sim \mathcal{N}(0,1)} [\phi'(g)]$$
$$\nu = \mathop{\mathbb{E}}_{g \sim \mathcal{N}(0,1)} [g \phi'(g)] \sqrt{\frac{\text{Tr}[\boldsymbol{\Sigma}^2]}{d}}$$
$$\vartheta_0 = \mathop{\mathbb{E}}_{g \sim \mathcal{N}(0,1)} [\phi(g)]$$
$$\vartheta_1 = \mathop{\mathbb{E}}_{g \sim \mathcal{N}(0,1)} [g \phi'(g)]$$
$$\vartheta_2 = \mathop{\mathbb{E}}_{g \sim \mathcal{N}(0,1)} [(\frac{1}{2} g^3 - g) \phi'(g)]$$

Note that $\boldsymbol{\psi}(\boldsymbol{x})$ consists of a scaled version of the data, a bias term, and a term that depends on the norm of the example. We will adopt the notation $f^{lin}(\boldsymbol{x}; \boldsymbol{\beta}) = \boldsymbol{\psi}(\boldsymbol{x})^\top \boldsymbol{\beta}$ for the linear model.

We can now formally state Assumption A.1:

**Assumption A.6** (formal version of Assumption A.1)**.** Suppose that Theorem A.5 holds. Then with high probability, for all such $t$ the following also holds for all $c \in C$ and for all $s \in \mathcal{A}$:

$$|f^{lin}(\boldsymbol{\beta}_t, \boldsymbol{v}_c) - f(\boldsymbol{w}_t, \boldsymbol{v}_c)| \leq O(\frac{\eta t}{d^{1+\Omega(\alpha)}}),$$
$$|f^{lin}(\boldsymbol{\beta}_t, \boldsymbol{v}_s) - f(\boldsymbol{w}_t, \boldsymbol{v}_s)| \leq O(\frac{\eta t}{d^{1+\Omega(\alpha)}}).$$

We will assume the former holds in the proof of the following theorems, although as we will see the assumption is unnecessary for Corollary 4.2.

# B PROOF FOR THEOREMS

## B.1 Notation

For the analysis, we split $\boldsymbol{\beta}$ into its components corresponding to the data, bias and norm parts of $\boldsymbol{\psi}$; that is $\boldsymbol{\beta} = \begin{pmatrix} \boldsymbol{\beta}' \\ \beta_{bias} \\ \beta_{norm} \end{pmatrix}$ for $\boldsymbol{\beta}' \in \mathbb{R}^d, \beta_{bias} \in \mathbb{R}, \beta_{norm} \in \mathbb{R}$. We use the inner product between $\boldsymbol{\beta}'$ and a feature $\boldsymbol{v}$ to understand how well the linear model learns a feature $\boldsymbol{v} \in \mathbb{R}^d$. With slight abuse of notation, we will simply write $\langle \boldsymbol{\beta}, \boldsymbol{v} \rangle$ to mean $\langle \boldsymbol{\beta}', \boldsymbol{v} \rangle$.

We also define the matrix $\boldsymbol{\Phi} = \begin{bmatrix} \phi_1 & \dots & \phi_n \end{bmatrix}^\top$.

## B.2 Proof of Theorem 4.1 and Corollary 4.2

**Theorem 4.1.** *Let $\alpha \in (0, \frac{1}{4})$ be a fixed constant. Suppose the number of training samples $n$ and the network width $m$ satisfy $n \gtrsim d^{1+\alpha}$ and $m \gtrsim d^{1+\alpha}$. Let $n_c$ be the number of examples in class $c$, and $n_{c,s} = |g_{c,s}|$ be the size of group $g_{c,s}$ with label $c$ and spurious feature $\boldsymbol{v}_s \in \mathcal{A}$. Then, under the setting of Sec. 3 there exist a constant $\nu_1 > 0$, such that with high probability, for all $0 \leq t \leq \nu_1 \cdot \sqrt{\frac{d^{1-\alpha}}{\eta}}$, the contribution of the core and spurious features to the network output can be quantified as follows:*

$$f(\boldsymbol{v}_c; \boldsymbol{W}_t, \boldsymbol{z}_t) = \frac{2\eta\zeta^2 c \|\boldsymbol{v}_c\|^2 t}{d} \left( \frac{n_c}{n} \pm O(d^{-\Omega(\alpha)}) \right), \tag{5}$$

$$f(\boldsymbol{v}_s; \boldsymbol{W}_t, \boldsymbol{z}_t) = \frac{2\eta\zeta^2 c \|\boldsymbol{v}_s\|^2 t}{d} \Bigg( \frac{n_{c,s} - n_{c',s}}{n} \tag{6}$$

$$\pm O(d^{-\Omega(\alpha)}) \Bigg),$$

*where $c' = C \backslash c$, and $\zeta$ is the expected gradient of activation functions at random initialization.*

**Corollary 4.2 (Separability of majority and minority groups).** *Suppose that for all classes, a majority group has at least $K$ examples and a minority group has at most $k$ examples. Then, under the assumptions of Theorem 4.1, examples in the majority and minority groups are nearly separable with high probability based on the model's output, early in training. That is, for all $0 \leq t \leq \nu_1 \cdot \sqrt{\frac{d^{1-\alpha}}{\eta}}$, with high probability, the following holds for at least $1 - O(d^{-\Omega(\alpha)})$ fraction of the training examples $\boldsymbol{x}_i$ in group $g_{c,s}$:*

*If $g_{c,s}$ is in a majority group in class $c = 1$:*

$$f(\boldsymbol{x}_i; \boldsymbol{W}_t, \boldsymbol{z}_t) \geq \frac{2\eta\zeta^2 t}{d} \Bigg( \frac{\|\boldsymbol{v}_s\|^2 (K - k)}{n} \tag{7}$$

$$+ \xi \pm O(d^{-\Omega(\alpha)}) \Bigg) + \rho(t, \phi, \Sigma),$$

*If $g_{c,s}$ is in a minority group in class $c = 1$, but $g_{c',s}$ is a majority group in class $c' = -1$:*

$$f(\boldsymbol{x}_i; \boldsymbol{W}_t, \boldsymbol{z}_t) \leq \frac{2\eta\zeta^2 t}{d} \Bigg( - \frac{\|\boldsymbol{v}_s\|^2 (K - k)}{n} \tag{8}$$

$$+ \xi \pm O(d^{-\Omega(\alpha)}) \Bigg) + \rho(t, \phi, \Sigma),$$

*where $\rho$ is constant for all examples in the same class, $\xi \sim \mathcal{N}(0, \kappa)$ with $\kappa = \frac{1}{n}(\sum_c n_c^2 \sigma_c^2 \|\boldsymbol{v}_c\|^2)^{1/2} + \frac{1}{n}(\sum_s (n_{c,s} - n_{c',s})^2 \sigma_s^2 \|\boldsymbol{v}_s\|^2)^{1/2}$ is the total effect of noise on the model.*

*Analogous statements holds for the class $c = -1$ by changing the sign and direction of the inequality.*

As in Hu et al. (2020), we will conduct our analysis under the high probability events that $\|\boldsymbol{\Psi}^\top \boldsymbol{\Psi}\| = O(\frac{n}{d})$ and for all training data $\boldsymbol{x}$, $\frac{\|\boldsymbol{x}\|}{\sqrt{d}} = 1 \pm O(\sqrt{\frac{\log n}{d}})$.

Starting from the rule of gradient descent

$$\begin{aligned}\beta(t+1) &= \beta(t) - \frac{\eta}{n}\boldsymbol{\Psi}^\top(\boldsymbol{\Psi}\beta(t) - \boldsymbol{y})\\ &= \left(I - \frac{\eta}{n}\boldsymbol{\Psi}^\top\boldsymbol{\Psi}\right)\beta(t) + \frac{\eta}{n}\boldsymbol{\Psi}^\top\boldsymbol{y}\end{aligned}$$

Let $\boldsymbol{A} = \boldsymbol{I} - \frac{\eta}{n}\boldsymbol{\Psi}^\top\boldsymbol{\Psi}, \boldsymbol{b} = \frac{\eta}{n}\boldsymbol{\Psi}^\top\boldsymbol{y}$. Also, $\boldsymbol{A}$ can be diagonalized as $\boldsymbol{A} = \boldsymbol{V}\boldsymbol{D}\boldsymbol{V}^\top$. Since $\|\boldsymbol{\Psi}^\top\boldsymbol{\Psi}\| = O(\frac{n}{d})$, the eigenvalues of $\boldsymbol{A}$, call them $\lambda_1, \ldots, \lambda_d$, are of order $1 - O(\frac{\eta}{d})$. For $t \geq 1$, the previous recurrence relation admits the solution

$$\begin{aligned}\beta(t) &= (\boldsymbol{I} + \boldsymbol{A} + \cdots + \boldsymbol{A}^{t-1})\boldsymbol{b}\\ &= \boldsymbol{V}(\boldsymbol{I} + \boldsymbol{D} + \cdots + \boldsymbol{D}^{t-1})\boldsymbol{V}^\top\boldsymbol{b}\end{aligned}$$

The eigenvalues of $\boldsymbol{I}+\boldsymbol{D}+\cdots+\boldsymbol{D}^{t-1}$ is a geometric series $\frac{1-\lambda_i^t}{1-\lambda_i}$, where $\lambda_i$ are the eigenvalues of $\boldsymbol{D}$. By the binomial theorem,

$$\begin{aligned}1 + \lambda_i + \cdots + \lambda_i^{t-1} &= \frac{1-\lambda_i^t}{1-\lambda_i}\\ &= \frac{1 - (1 - t(1-\lambda_i) + O(t^2(1-\lambda_i)^2))}{1-\lambda_i}\\ &= t + O(t^2(1-\lambda))\end{aligned}$$

When $t = O(\sqrt{\frac{d^{1-\alpha}}{\eta}})$, the expression simplifies to $t+O(d^{-\alpha/2})$. Thus we can approximate $\boldsymbol{I}+\boldsymbol{D}+\cdots+\boldsymbol{D}^{t-1} = t\boldsymbol{I}+\boldsymbol{\Delta}$, where $\|\boldsymbol{\Delta}\| = O(d^{-\frac{\alpha}{2}})$. Then

$$\beta(t) = \boldsymbol{V}(t\boldsymbol{I} + \boldsymbol{\Delta})\boldsymbol{V}^\top\boldsymbol{b} = t\boldsymbol{b} + \boldsymbol{\Delta}_1\boldsymbol{b}$$

where $\boldsymbol{\Delta}_1 = \boldsymbol{V}\boldsymbol{\Delta}\boldsymbol{V}^\top$ also satisfies $\|\boldsymbol{\Delta}\| = O(d^{-\frac{\alpha}{2}})$.

From here we may calculate the following: the alignment of $\boldsymbol{\beta}$ with a core feature $\boldsymbol{v}_c$ is

$$\begin{aligned}\langle \boldsymbol{v}_c, \boldsymbol{\beta}\rangle &= \left\langle \boldsymbol{v}_c, \frac{\eta t}{n}\boldsymbol{\Psi}^\top\boldsymbol{y} + \boldsymbol{\Delta}_1\frac{\eta}{n}\boldsymbol{\Psi}^\top\boldsymbol{y}\right\rangle & (12)\\ &= \frac{\eta}{n}\sum_{i=1}^{n}\langle \boldsymbol{v}_c, t y_i \boldsymbol{\psi}_i + \boldsymbol{\Delta}_1 y_i \boldsymbol{\psi}_i\rangle & (13)\\ &= \sqrt{\frac{2}{d}}\frac{\eta\zeta c\|\boldsymbol{v}_c\|}{n}(t \pm O(d^{-\frac{\alpha}{2}}))(\|\boldsymbol{v}_c\|n_c \pm O(\sigma_c\sqrt{n})) & (14)\\ &= \sqrt{\frac{2}{d}}\eta\zeta c\|\boldsymbol{v}_c\|^2 t\left(\frac{n_c}{n} \pm O(d^{-\Omega(\alpha)})\right) & (15)\end{aligned}$$

and the alignment with a spurious feature $\boldsymbol{v}_s$ is

$$\begin{aligned}\langle \boldsymbol{v}_s, \boldsymbol{\beta}\rangle &= \left\langle \boldsymbol{v}_s, \frac{\eta t}{n}\boldsymbol{\Psi}^\top\boldsymbol{y} + \boldsymbol{\Delta}_1\frac{\eta}{n}\boldsymbol{\Psi}^\top\boldsymbol{y}\right\rangle & (16)\\ &= \frac{\eta}{n}\sum_{i=1}^{n}\langle \boldsymbol{v}_s, t y_i \boldsymbol{\psi}_i + \boldsymbol{\Delta}_1 y_i \boldsymbol{\psi}_i\rangle & (17)\\ &= \sqrt{\frac{2}{d}}\frac{\eta\zeta c\|\boldsymbol{v}_s\|}{n}(t \pm O(d^{-\frac{\alpha}{2}}))(\|\boldsymbol{v}_s\|(n_{c,s} - n_{c',s}) \pm O(\sigma_s\sqrt{n})) & (18)\\ &= \sqrt{\frac{2}{d}}\eta\zeta c\|\boldsymbol{v}_s\|^2 t\left(\frac{n_{c,s} - n_{c',s}}{n} \pm O(d^{-\Omega(\alpha)})\right) & (19)\end{aligned}$$

Eq. 12, 16 hold by substituting $\beta = \frac{\eta t}{n}\boldsymbol{\Psi}^\top\boldsymbol{y} + \boldsymbol{\Delta}_1\boldsymbol{b}$ error derived earlier and considering the inner product of every column of $\boldsymbol{\Psi}^\top$ with the core/spurious feature. The effect of the noise is captured by the $O(\sigma\sqrt{n})$ terms, following standard concentration inequalities, and we used the fact that $\frac{1}{\sqrt{n}} = O(d^{-\Omega(\alpha)})$.

In addition, we calculate that

$$\beta_{norm}(t) = (tI + \mathbf{\Delta}_1) \sum_{i=1}^{n} y_i \left( \vartheta_0 + \vartheta_1 (\frac{\|\boldsymbol{x}_i\|}{\sqrt{d}} - 1) + \vartheta_2 (\frac{\|\boldsymbol{x}_i\|}{\sqrt{d}} - 1)^2 \right) = O(\frac{\eta t}{\sqrt{n}})$$

Now the result transfers to the full neural network under Assumption A.6, namely

$$\begin{aligned} f(\boldsymbol{v}_c; \boldsymbol{W}_t, \boldsymbol{z}_t) &= f^{lin}(\boldsymbol{\beta}_t, \boldsymbol{v}_c) \pm O(\frac{\eta t}{d^{1+\Omega(\alpha)}}) && (20) \\ &= \frac{2\eta\zeta^2 c \|\boldsymbol{v}_c\|^2 t}{d} \left( \frac{n_c}{n} \pm O(d^{-\Omega(\alpha)}) \right), && (21) \\ f(\boldsymbol{v}_s; \boldsymbol{W}_t, \boldsymbol{z}_t) &= f^{lin}(\boldsymbol{\beta}_t, \boldsymbol{v}_s) \pm O(\frac{\eta t}{d^{1+\Omega(\alpha)}}) && (22) \\ &= \frac{2\eta\zeta^2 c \|\boldsymbol{v}_s\|^2 t}{d} \left( \frac{n_{c,s} - n_{c',s}}{n} \pm O(d^{-\Omega(\alpha)}) \right), && (23) \end{aligned}$$

This proves Theorem 4.1.

Then for the predictions at time $t$ for an example in class $c = 1$, group $g_{1,s}$:

$$\begin{aligned} \boldsymbol{\psi}(\boldsymbol{x})^\top \boldsymbol{\beta}(t) &= \sqrt{\frac{2}{d}} \zeta \boldsymbol{x}^\top \boldsymbol{\beta}' + \sqrt{\frac{3}{2d}} \nu \beta_{bias}(t) + \beta_{norm}(t) \left( \vartheta_0 + \vartheta_1 (\frac{\|\boldsymbol{x}\|}{\sqrt{d}} - 1) + \vartheta_2 (\frac{\|\boldsymbol{x}\|}{\sqrt{d}} - 1)^2 \right) \\ &= \sqrt{\frac{2}{d}} \zeta (\boldsymbol{v}_1 + \boldsymbol{v}_s + \boldsymbol{\xi})^\top \boldsymbol{\beta}' + \sqrt{\frac{3}{2d}} \nu \beta_{bias}(t) + \vartheta_0 \beta_{norm}(t) \pm O\left( \eta t \sqrt{\frac{\log n}{nd}} \right) \end{aligned}$$

We have a few cases

1. $g_{1,k}$ is a majority group. In this case

$$\begin{aligned} \boldsymbol{\psi}(\boldsymbol{x})^\top \boldsymbol{\beta}(t) \geq{} & \frac{2\eta\zeta^2 t}{d} \left( \frac{n_1 \|\boldsymbol{v}_c\|^2}{n} + \frac{\|\boldsymbol{v}_s\|^2 (K-k)}{n} + \left\langle \boldsymbol{\xi}, \frac{1}{n} \boldsymbol{X}^\top \boldsymbol{y} \right\rangle \pm O(d^{-\Omega(\alpha)}) \right) \\ & + \sqrt{\frac{3}{2d}} \nu \beta_{bias}(t) + \vartheta_0 \beta_{norm}(t) \pm O\left( \eta t \sqrt{\frac{\log n}{nd}} \right) \end{aligned}$$

2. $g_{1,k}$ is a minority group and $g_{-1,k}$ is a majority group. In this case

$$\begin{aligned} \boldsymbol{\psi}(\boldsymbol{x})^\top \boldsymbol{\beta}(t) \leq{} & \frac{2\eta\zeta^2 t}{d} \left( \frac{n_1 \|\boldsymbol{v}_c\|^2}{n} - \frac{\|\boldsymbol{v}_s\|^2 (K-k)}{n} + \left\langle \boldsymbol{\xi}, \frac{1}{n} \boldsymbol{X}^\top \boldsymbol{y} \right\rangle \pm O(d^{-\Omega(\alpha)}) \right) \\ & + \sqrt{\frac{3}{2d}} \nu \beta_{bias}(t) + \vartheta_0 \beta_{norm}(t) \pm O\left( \eta t \sqrt{\frac{\log n}{nd}} \right) \end{aligned}$$

3. $g_{1,k}$ is such that no majority groups have the spurious feature. In this case

$$\begin{aligned} \boldsymbol{\psi}(\boldsymbol{x})^\top \boldsymbol{\beta}(t) ={} & \frac{2\eta\zeta^2 t}{d} \left( \frac{n_1 \|\boldsymbol{v}_c\|^2}{n} + \frac{\|\boldsymbol{v}_s\|^2 \tilde{k}}{n} + \left\langle \boldsymbol{\xi}, \frac{1}{n} \boldsymbol{X}^\top \boldsymbol{y} \right\rangle \pm O(d^{-\Omega(\alpha)}) \right) \\ & + \sqrt{\frac{3}{2d}} \nu \beta_{bias}(t) + \vartheta_0 \beta_{norm}(t) \pm O\left( \eta t \sqrt{\frac{\log n}{nd}} \right), \qquad |\tilde{k}| \leq k \end{aligned}$$

Now

$$\begin{aligned} \left\langle \boldsymbol{\xi}, \frac{1}{n} \boldsymbol{X}^\top \boldsymbol{y} \right\rangle &= \sum_{c \in \{\pm 1\}} \frac{\|\boldsymbol{v}_c\| n_c}{n} \langle \boldsymbol{\xi}, \boldsymbol{v}_c \rangle + \sum_{s} \frac{\|\boldsymbol{v}_s\| (n_{1,s} - n_{-1,s})}{n} \langle \boldsymbol{\xi}, \boldsymbol{v}_s \rangle + \left\langle \boldsymbol{\xi}, \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{\xi}_i y_i \right\rangle && (24) \\ &= \sum_{c \in \{\pm 1\}} \frac{\|\boldsymbol{v}_c\| n_c}{n} \langle \boldsymbol{\xi}, \boldsymbol{v}_c \rangle + \sum_{s} \frac{\|\boldsymbol{v}_s\| (n_{1,s} - n_{-1,s})}{n} \langle \boldsymbol{\xi}, \boldsymbol{v}_s \rangle \pm O\left( \sqrt{\frac{d}{n}} \right) && (25) \\ &\sim \mathcal{N}(0, \kappa) \pm O(d^{-\Omega(\alpha)}) && (26) \end{aligned}$$

Finally, observe that $O\left(\eta t\sqrt{\frac{\log n}{nd}}\right) = O(d^{-1-\Omega(\alpha)})$. Combining all these results and setting $\rho_1 = \frac{2\eta\zeta^2 ct}{d}, \rho_2 = \frac{\rho_1 n_1 \|\boldsymbol{v}_c\|^2}{n} + \sqrt{\frac{3}{2d}}\nu\beta_{bias}(t) + \vartheta_0\beta_{norm}(t)$ shows Corollary 4.2 when looking at the prediction of the linear model. Recall that Hu et al. (2020) showed that the average squared error in predictions between the linear model and the full neural network is $O(\frac{\eta^2 t^2}{d^{2+\Omega(\alpha)}})$. Then by Markov's inequality, we can guarantee that the predictions of the linear model differ by at most $O(\frac{\eta t}{d^{1+\Omega(\alpha)}})$ for at least $1 - O(d^{-\Omega(\alpha)})$ proportion of the examples. This error can be factored into the existing error term. Hence the result holds for the full neural network.

We can apply the same argument for the class $c'$. Thus Corollary 4.2 is proven.

Notably, Corollary 4.2 only depends on the closeness of the neural network and the initial linear model on the training data, hence does not rely on Assumption A.6.

### B.3 Proof of Theorem 4.3

**Theorem 4.3.** *Under the assumptions of Theorem 4.1, if the classes are balanced, and the total size of the minority groups in class $c$ is small, i.e., $O(n^{1-\gamma})$ for some $\gamma > 0$, then there exists a constant $\nu_2 > 0$ such that at $T = \nu_2 \cdot \frac{d \log d}{\eta}$, for an example $\boldsymbol{x}_i$ in a majority group $g_{c,s}$, the contribution of the core feature to the model's output is at most:*

$$|f(\boldsymbol{v}_c; \boldsymbol{W}_T, \boldsymbol{z}_T)| \leq \left(\frac{\sqrt{2}R_s}{R_c} + O(n^{-\gamma} + d^{-\Omega(\alpha)})\right). \tag{9}$$

*In particular if $\min\{R_c, 1\} \gg R_s$, then the model's output is mostly indicated by the spurious feature instead of the core feature:*

$$|f(\boldsymbol{v}_s; \boldsymbol{W}_T, \boldsymbol{z}_T)| \gg |f(\boldsymbol{v}_c; \boldsymbol{W}_T, \boldsymbol{z}_T)|. \tag{10}$$

Let $g_{maj}$ be the total number of majority groups among all classes. Note that by the definition of majority groups, $g_{maj}$ is at most the number of classes, namely 2 in the given analysis.

Since the classes are balanced with labels ±1, it is not hard to see that the bias term in the weights will always be zero, hence we may as well assume that we do not have the bias term. Abusing notation, we will still denote quantities by the same symbol, even though now the bias term has been removed.

First consider a model $\tilde{f} = \boldsymbol{\psi}^\top\tilde{\boldsymbol{\beta}}$ trained on the dataset $\mathcal{D}_{\text{maj}}$, which only contains examples from the majority groups, and the norm term of $\boldsymbol{\psi}$ is only the constant term $\vartheta_0$ instead of $\vartheta_0 + \vartheta_1(\frac{\|\boldsymbol{x}\|}{\sqrt{d}} - 1) + \vartheta_2(\frac{\|\boldsymbol{x}\|}{\sqrt{d}} - 1)^2$. Further, assume $\mathcal{D}_{\text{maj}}$ has infinitely many examples so that the noise perfectly matches the underlying distribution. We prove the results in this simplified setting then extend the result using matrix perturbations.

We have

$$\mathcal{L} = \frac{1}{2}\mathop{\mathbb{E}}_{\mathcal{D}_{\text{maj}}}[(\boldsymbol{\psi}_i^\top\tilde{\boldsymbol{\beta}} - y_i)^2]$$
$$\nabla\mathcal{L} = \mathop{\mathbb{E}}_{\mathcal{D}_{\text{maj}}}[(\boldsymbol{\psi}_i^\top\tilde{\boldsymbol{\beta}} - y_i)\boldsymbol{\psi}_i]$$

and the optimal $\tilde{\boldsymbol{\beta}}_*$ satisfies

$$\tilde{\boldsymbol{\beta}}_* = \left(\mathop{\mathbb{E}}_{\mathcal{D}_{\text{maj}}}[\boldsymbol{\psi}_i\boldsymbol{\psi}_i^\top]\right)^\dagger \mathop{\mathbb{E}}_{\mathcal{D}_{\text{maj}}}[y_i\boldsymbol{\psi}_i]$$

where † represents the Moore-Penrose pseudo-inverse.

Since the noise is symmetrical with respect to the classes, the bias and constant norm terms of $\boldsymbol{\beta}$ must be zero Formally, the first order condition implies

$$\mathop{\mathbb{E}}_{\mathcal{D}_{\text{maj}}}[\boldsymbol{\psi}_i\boldsymbol{\psi}_i^\top]\tilde{\boldsymbol{\beta}}_* = \mathop{\mathbb{E}}_{\mathcal{D}_{\text{maj}}}[y_i\boldsymbol{\psi}_i]$$

As shorthand, denote $\alpha = \begin{bmatrix}\psi_{bias}\\ \vartheta_0\end{bmatrix}^\top \begin{bmatrix}\tilde{\beta}^*_{bias}\\ \tilde{\beta}^*_{norm}\end{bmatrix}$ Decomposing the above vector equation based on the bias and constant norm terms versus the others, we have

$$\mathop{\mathbb{E}}_{\mathcal{D}_{\text{maj}}}\left[\begin{bmatrix}\psi_{bias}\\ \vartheta_0\end{bmatrix}\left(\sqrt{\frac{2}{d}}\zeta \boldsymbol{x}_i\right)^\top \tilde{\boldsymbol{\beta}}'\right] + \begin{bmatrix}\psi_{bias}\\ \vartheta_0\end{bmatrix}\alpha = \begin{bmatrix}0\\ 0\end{bmatrix} \tag{27}$$

$$\mathop{\mathbb{E}}_{\mathcal{D}_{\text{maj}}}\left[\left(\sqrt{\frac{2}{d}}\zeta \boldsymbol{x}_i\right)^\top \tilde{\boldsymbol{\beta}}'\right] + \alpha = 0 \tag{28}$$

The remaining coordinates give the equation

$$\mathop{\mathbb{E}}_{\mathcal{D}_{\text{maj}}}\left[\left(\sqrt{\frac{2}{d}}\zeta \boldsymbol{x}_i\right)\left(\sqrt{\frac{2}{d}}\zeta \boldsymbol{x}_i\right)^\top \tilde{\boldsymbol{\beta}}'\right] + \mathop{\mathbb{E}}_{\mathcal{D}_{\text{maj}}}\left[\left(\sqrt{\frac{2}{d}}\zeta \boldsymbol{x}_i\right)\alpha\right] = \mathop{\mathbb{E}}_{\mathcal{D}_{\text{maj}}}\left[y_i\sqrt{\frac{2}{d}}\zeta \boldsymbol{x}_i\right] \tag{29}$$

Set $\boldsymbol{z} = \sum_c \frac{\boldsymbol{v}_c}{\|v_c\|^2}$. Observe that for all $\boldsymbol{x}_i$,

$$\boldsymbol{z}^\top \boldsymbol{x}_i = 1 + \boldsymbol{z}^\top \boldsymbol{\xi}_i$$

Taking the inner product of the latter equation 29 with $\boldsymbol{z}$ gives

$$\mathop{\mathbb{E}}_{\mathcal{D}_{\text{maj}}}\left[\sqrt{\frac{2}{d}}\zeta(1+\boldsymbol{z}^\top\boldsymbol{\xi}_i)\left(\sqrt{\frac{2}{d}}\zeta \boldsymbol{x}_i\right)^\top \tilde{\boldsymbol{\beta}}'\right] + \mathop{\mathbb{E}}_{\mathcal{D}_{\text{maj}}}\left[\sqrt{\frac{2}{d}}\zeta(1+\boldsymbol{z}^\top\boldsymbol{\xi}_i)\alpha\right] = \mathop{\mathbb{E}}_{\mathcal{D}_{\text{maj}}}\left[y_i\sqrt{\frac{2}{d}}\zeta(1+\boldsymbol{z}^\top\boldsymbol{\xi}_i)\right]$$

$$\mathop{\mathbb{E}}_{\mathcal{D}_{\text{maj}}}\left[\frac{2}{d}\zeta^2(\boldsymbol{x}_i^\top\tilde{\boldsymbol{\beta}}' + \boldsymbol{z}^\top\boldsymbol{\xi}_i\boldsymbol{\xi}_i^\top\tilde{\boldsymbol{\beta}}')\right] + \sqrt{\frac{2}{d}}\zeta\alpha = 0$$

$$\mathop{\mathbb{E}}_{\mathcal{D}_{\text{maj}}}\left[\frac{2}{d}\zeta^2\boldsymbol{x}_i^\top\tilde{\boldsymbol{\beta}}'\right] + \frac{2}{d}\zeta^2\sum_c \frac{\sigma_c^2}{\|v_c\|^2}\boldsymbol{v}_c^\top\tilde{\boldsymbol{\beta}}' + \sqrt{\frac{2}{d}}\zeta\alpha = 0$$

$$\mathop{\mathbb{E}}_{\mathcal{D}_{\text{maj}}}\left[\sqrt{\frac{2}{d}}\zeta\boldsymbol{x}_i^\top\tilde{\boldsymbol{\beta}}'\right] + \sqrt{\frac{2}{d}}\zeta R_c^2\sum_c \boldsymbol{v}_c^\top\tilde{\boldsymbol{\beta}}' + \alpha = 0$$

Combined with equation 28, we conclude that $\sum_c \boldsymbol{v}_c^\top\tilde{\boldsymbol{\beta}}' = 0$. But since the classes are balanced, this implies that $\mathbb{E}_{\mathcal{D}_{\text{maj}}}[\boldsymbol{v}_{c_i}^\top\tilde{\boldsymbol{\beta}}'] = 0$. A similar argument shows that $\mathbb{E}_{\mathcal{D}_{\text{maj}}}[\boldsymbol{v}_{s_i}^\top\tilde{\boldsymbol{\beta}}'] = 0$. We conclude that $\mathbb{E}_{\mathcal{D}_{\text{maj}}}[\boldsymbol{x}_i^\top\tilde{\boldsymbol{\beta}}'] = 0$, hence $\alpha = 0$. Now since the solution $\tilde{\boldsymbol{\beta}}^*$ lies in the span of the data, we must have

$$\left\|\begin{bmatrix}\tilde{\beta}^*_{bias}\\ \tilde{\beta}^*_{norm}\end{bmatrix}\right\|^2 = \alpha = 0$$

We conclude that $\tilde{\beta}^*_{bias} = \tilde{\beta}^*_{norm} = \alpha = 0$, as claimed.

Thus the loss becomes

$$\mathcal{L} = \frac{1}{2}\mathop{\mathbb{E}}_{(\boldsymbol{x}_i,y_i)\sim\mathcal{D}_{\text{maj}}}\left[(\sqrt{\frac{2}{d}}\zeta\boldsymbol{x}_i^\top\tilde{\boldsymbol{\beta}}' - y_i)^2\right] \tag{30}$$

$$= \frac{1}{2}\mathop{\mathbb{E}}_{\mathcal{D}_{\text{maj}}}\left[(\sqrt{\frac{2}{d}}\zeta(\boldsymbol{v}_{c_i} + \boldsymbol{v}_{s_i} + \boldsymbol{\xi}_i)^\top\tilde{\boldsymbol{\beta}}' - y_i)^2\right] \tag{31}$$

$$= \frac{1}{2}\mathop{\mathbb{E}}_{\mathcal{D}_{\text{maj}}}\left[(\sqrt{\frac{2}{d}}\zeta(\boldsymbol{v}_{c_i} + \boldsymbol{v}_{s_i})^\top\tilde{\boldsymbol{\beta}}' - y_i)^2 + (\sqrt{\frac{2}{d}}\zeta\boldsymbol{\xi}_i^\top\tilde{\boldsymbol{\beta}}')^2\right] \tag{32}$$

$$= \frac{1}{2}\mathop{\mathbb{E}}_{\mathcal{D}_{\text{maj}}}\left[(\sqrt{\frac{2}{d}}\zeta(\boldsymbol{v}_{c_i} + \boldsymbol{v}_{s_i})^\top\tilde{\boldsymbol{\beta}}' - y_i)^2\right] + \frac{\zeta^2}{d}\tilde{\boldsymbol{\beta}}'^\top\boldsymbol{\Sigma}_\xi\tilde{\boldsymbol{\beta}}' \tag{33}$$

Consider the model $\boldsymbol{\beta}_s$ which only learns the spurious features of majority groups

$$\boldsymbol{\beta}'_s = \sqrt{\frac{d}{2}}\frac{1}{\zeta}\sum_{g_{c,s}\text{is a majority group}}\frac{c\boldsymbol{v}_s}{\|\boldsymbol{v}_s\|^2}.$$

Note that for any example in a majority group, $(\boldsymbol{v}_{c_i} + \boldsymbol{v}_{s_i})^\top \boldsymbol{\beta}'_s - y_i = 0$. Thus

$$\begin{aligned}
\mathcal{L} &= \frac{\zeta^2}{d} \tilde{\boldsymbol{\beta}}'^\top \boldsymbol{\Sigma}_\xi \tilde{\boldsymbol{\beta}}' \\
&= \sum_{\boldsymbol{v}_s \text{is spurious}} \frac{\sigma_s^2}{2\|\boldsymbol{v}_s\|^2} \\
&\leq \frac{g_{\mathrm{maj}} R^2}{2}
\end{aligned}$$

The loss for the optimal model must be smaller. But the loss due to the last term in Eq. (33) along a core feature alone is

$$\frac{\zeta^2 \sigma_c^2}{\|\boldsymbol{v}_c\|^2 d} \left\langle \boldsymbol{v}_c, \boldsymbol{\beta}'_* \right\rangle^2 \leq \frac{g_{\mathrm{maj}} R_s^2}{2}$$

Rearranging gives

$$\left\langle \boldsymbol{v}_c, \boldsymbol{\beta}'_* \right\rangle^2 \leq \frac{d g_{\mathrm{maj}} R^2 \|\boldsymbol{v}_c\|^2}{2\zeta^2 \sigma_c^2} \tag{34}$$

Now consider the loss from the first term in Eq. (33) due to a majority group. It must be at least

$$\begin{aligned}
\frac{K}{n}\left(1 - \sqrt{\frac{2}{d}} \zeta \left\langle \boldsymbol{v}_s, \boldsymbol{\beta}'_* \right\rangle - \frac{\sqrt{g_{\mathrm{maj}}} R_s \|\boldsymbol{v}_c\|}{\sigma_c}\right)^2 &\leq \frac{g_{\mathrm{maj}} R_s^2}{2} \\
1 - \sqrt{\frac{2}{d}} \zeta \left\langle \boldsymbol{v}_s, \boldsymbol{\beta}'_* \right\rangle - \frac{\sqrt{g_{\mathrm{maj}}} R_s \|\boldsymbol{v}_c\|}{\sigma_c} &\leq \sqrt{\frac{n g_{\mathrm{maj}} R_s^2}{2K}} \\
1 - \sqrt{g_{\mathrm{maj}}} R_s \left(\frac{\|\boldsymbol{v}_c\|}{\sigma_c} + \sqrt{\frac{n}{2K}}\right) &\leq \sqrt{\frac{2}{d}} \zeta \left\langle \boldsymbol{v}_s, \boldsymbol{\beta}'_* \right\rangle
\end{aligned}$$

Note that $\sqrt{\frac{n}{2K}} \leq \sqrt{\frac{g_{maj}}{2}}$. Now if we have $R_s$ sufficiently smaller than $\frac{\sigma_c}{\sqrt{g_{maj}} \|\boldsymbol{v}_c\|}$ and $\frac{2}{g_{\mathrm{maj}}}$, we can guarantee that the RHS is at least some constant less than 1, say $\frac{1}{\sqrt{2}}$. In this case, we have

$$\langle \boldsymbol{v}_s, \boldsymbol{\beta}_* \rangle^2 \geq \frac{d}{4\zeta^2} \tag{35}$$

Under these assumptions, it is clear from Eq. (34) that we will also have

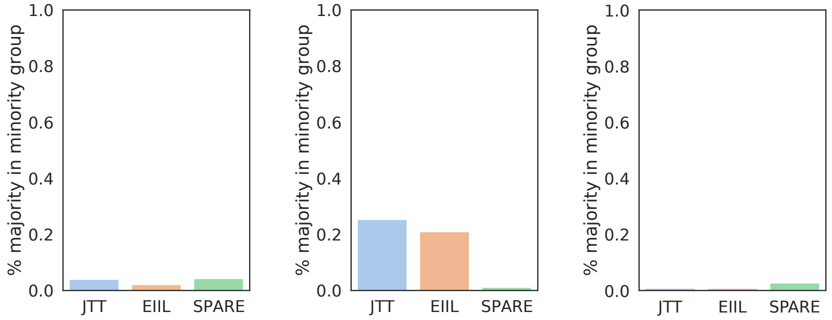
$$\frac{d}{4\zeta^2} \gg \langle \boldsymbol{v}_c, \boldsymbol{\beta}_* \rangle^2 \tag{36}$$

Now we return to the original dataset, which contains minority groups and only a finite number of examples. Again, we have

$$\boldsymbol{\beta}_* = (\boldsymbol{\Psi}^\top \boldsymbol{\Psi})^\dagger \boldsymbol{\Psi}^\top \boldsymbol{y}$$

Since we have removed the bias term, it is not hard to show that the matrix $\frac{1}{n}\boldsymbol{\Psi}^\top \boldsymbol{\Psi}$ has all eigenvalues of order $\Theta(\frac{1}{d})$. Now consider the norm of the difference $\|\frac{1}{n}\boldsymbol{\Psi}^\top \boldsymbol{\Psi} - \mathbb{E}_{\mathcal{D}_{\mathrm{maj}}}[\psi_i \psi_i^\top]\|$. With high probability, it will be of order $O(\frac{n_{\mathrm{mino}}}{nd} + \frac{1}{d}\sqrt{\frac{d}{n}} + \sqrt{\frac{\log n}{nd}}) = O(\frac{n^{-\gamma} + d^{-\Omega(\alpha)}}{d})$, where the first term corresponds to the inclusion of minority groups, the second term corresponds having a finite sample size, and the third term corresponds to using the true norm component. It follows that

$$\begin{aligned}
\left\| (\frac{1}{n}\boldsymbol{\Psi}^\top \boldsymbol{\Psi})^\dagger - (\mathop{\mathbb{E}}_{\mathcal{D}_{\mathrm{maj}}}[\psi_i \psi_i^\top])^\dagger \right\| &= O\left(d - \frac{d}{1 + O(n^{-\gamma} + d^{-\Omega(\alpha)}})\right) \\
&= O(d(n^{-\gamma} + d^{-\Omega(\alpha)}))
\end{aligned}$$



(a) Waterbirds (b) CelebA (c) UrbanCars

Figure 8: Fraction of majority examples inferred as minority. Spare not only identifies minority groups more accurately and correctly upweights them as evidenced in Fig. 3, but also does not identify a lot of majority examples as minority.

A similar argument shows that

$$\|\boldsymbol{\Psi}^\top \boldsymbol{y} - \mathop{\mathbb{E}}_{\mathcal{D}_{\text{maj}}}[y_i \psi_i]\| = O(d^{-\frac{1}{2}}(n^{-\gamma} + d^{-\Omega(\alpha)}))$$

Thus the change in alignment with a feature $\boldsymbol{v}$ is

$$\begin{aligned}\left\|\left\langle \tilde{\boldsymbol{\beta}}_*, \boldsymbol{v}\right\rangle - \langle \boldsymbol{\beta}_*, \boldsymbol{v}\rangle\right\| &= \left\|(\boldsymbol{\Psi}^\top\boldsymbol{\Psi})^\dagger \boldsymbol{\Psi}^\top \boldsymbol{y} - \left(\mathop{\mathbb{E}}_{\mathcal{D}_{\text{maj}}}[\psi_i\psi_i^\top]\right)^\dagger \mathop{\mathbb{E}}_{\mathcal{D}_{\text{maj}}}[y_i\psi_i]\right\| \|\boldsymbol{v}\| \\ &\leq \left\|\left((\boldsymbol{\Psi}^\top\boldsymbol{\Psi})^\dagger - \left(\mathop{\mathbb{E}}_{\mathcal{D}_{\text{maj}}}[\psi_i\psi_i^\top]\right)^\dagger\right)\boldsymbol{\Psi}^\top\boldsymbol{y}\right. \\ &\qquad \left. + \left(\mathop{\mathbb{E}}_{\mathcal{D}_{\text{maj}}}[\psi_i\psi_i^\top]\right)^\dagger (\boldsymbol{\Psi}^\top\boldsymbol{y} - \mathop{\mathbb{E}}_{\mathcal{D}_{\text{maj}}}[y_i\psi_i])\right\| \|\boldsymbol{v}\| \\ &\leq O\left(d(n^{-\gamma} + d^{-\Omega(\alpha)})(d^{-\frac{1}{2}}) + dd^{-\frac{1}{2}}(n^{-\gamma} + d^{-\Omega(\alpha)})\right) \\ &\leq O((n^{-\gamma} + d^{-\Omega(\alpha)})\sqrt{d})\end{aligned}$$

Replacing $g_{maj}$ with 2,and combining equations Eqs. (34), (35) and (37) gives

$$|\langle \boldsymbol{\beta}^*, \boldsymbol{v}_s\rangle| \geq \frac{\sqrt{d}}{2\zeta} \gg \sqrt{d}\left(\frac{R_s}{\zeta R_c} + O(n^{-\gamma} + d^{-\Omega(\alpha)})\right) \geq |\langle \boldsymbol{v}_c, \boldsymbol{\beta}^*\rangle|. \tag{37}$$

Then by Assumption A.6, we get

$$|f(\boldsymbol{v}_s; \boldsymbol{W}_T, \boldsymbol{z}_T)| \gg \frac{\sqrt{2}R_s}{R_c} + O(n^{-\gamma} + d^{-\Omega(\alpha)}) \geq |f(\boldsymbol{v}_c; \boldsymbol{W}_T, \boldsymbol{z}_T)|. \tag{38}$$

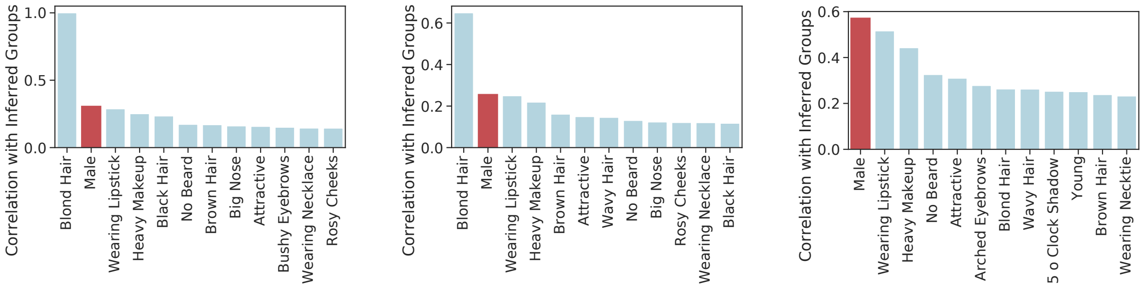
which proves the theorem.

## C Minimal Majority in Minority Groups by Spare

In Sec. 6.2, we showed Spare's ability to accurately identify minority groups with minimal minority identified in the majority groups. We note that, as evidenced in Fig. 8, the presence of majority examples within the minority groups identified by Spare is also low. This reduced rate minimizes the likelihood of incorrect upweighting.

## D Intercorrelation of Inferred Groups and Attributes in CelebA

To understand the attributes used to separate the groups, we measure the intercorrelation using Cramer's V (Cramér, 1999) between groups inferred by different group inference methods (JTT, EIIL, and Spare) and the



(a) JTT (b) EIIL (c) SPARE

Figure 9: Intercorrelation between the group inferred by state-of-the-art group inference methods (JTT, EIIL, Spare) and the attributes in the CelebA dataset, measured by Cramer's V (Cramér, 1999). The higher the value, the more likely the inferred groups can be completely determined by the attribute. JTT and EIIL mainly separate majority and minority based on the class attribute ("Blond Hair") while SPARE separates groups mainly based on the spurious attribute ("Male", colored in red).

attributes in the CelebA dataset. The metric allows us to measure the likelihood that a given attribute can completely determine the inferred groups.

As illustrated in Fig. 9, JTT and EIIL show a higher Cramer's V value for the class attribute "Blond Hair", indicating that their inferred groups are mainly based on this attribute. On the other hand, Spare (SPARE) exhibits a higher value for the spurious attribute "Male", colored in red, demonstrating that it more effectively separates groups based on the spurious attribute.

# E EXPERIMENTATION DETAILS

## E.1 Datasets

**CMNIST** We created a colored MNIST dataset with spurious correlations by using colors as spurious attributes following the settings in Zhang et al. (2022). First, we defined an image classification task with 5 classes by grouping consecutive digits (0 and 1, 2 and 3, 4 and 5, 6 and 7, 8 and 9) into the same class. From the train split, we randomly selected 50,000 examples as the training set, while the remaining 10,000 samples were used as the validation set. The test split follows the official test split of MNIST.

For each class $y_i$, we assigned a color $\boldsymbol{v}_s$ from a set of colors $\mathcal{A}$={#ff0000, #85ff00, #00fff3, #6e00ff, #ff0018} as the spurious attribute that highly correlates with this class, represented by their hex codes, to the foreground of a fraction $p_{corr}$ of the training examples. This fraction represents the majority group for class $y_i$. The stronger the spurious correlation between class $y_i$ and the spurious attribute $\boldsymbol{v}_s$, the higher the value of $p_{corr}$. The remaining $1 - p_{corr}$ training examples were randomly colored using a color selected from $\mathcal{A} \setminus \boldsymbol{v}_s$. In our experiments, we set $p_{corr} = 0.995$ to establish significant spurious correlations within the dataset.

**Waterbirds** is introduced by Sagawa et al. (2019) to study the spurious correlation between the background (land/water) and the foreground (landbird/waterbird) in image recognition. Species in Caltech-UCSD Birds-200-2011 (CUB-200-2011) dataset (Wah et al., 2011) are grouped into two classes, waterbirds and landbirds. All birds are then cut and pasted onto new background images, with waterbirds more likely to appear on water and landbirds having a higher probability on land. There are 4795 training examples in total, 3498 for landbirds with land background, 184 for landbirds with water background, 56 for waterbirds with land background, and 1057 for waterbirds with water background.

**CelebA** is a large-scale face attribute dataset comprised of photos of celebrities. Each image is annotated with 40 binary attributes, in which "blond hair" and "male" are commonly used for studying spurious correlations. Specifically, gender is considered a spurious feature for hair color classification. The smallest group is blond male.

**UrbanCars** The UrbanCars dataset is introduced by Li et al. (2023) to explore the impact of multiple spurious correlations in image classification. Each image features a car centrally placed against a natural scene background, accompanied by a co-occurring object to the right. The goal is to classify the car's body type while accounting for two spurious attributes: the background (BG) and the co-occurring object (CoObj), which are correlated with the target label. The labels share a binary space, consisting of two classes: urban and country. The dataset is partitioned into 8 groups based on various combinations of these labels. The training set manifests strong spurious correlations of 0.95 in strength for both BG and CoObj.

UrbanCars is assembled from multiple source datasets, including Stanford Cars (Krause et al., 2013) for the car objects and labels, Places (Zhou et al., 2018) for the backgrounds, and LVIS (Gupta et al., 2019) for the co-occurring objects. Backgrounds and co-occurring objects are selected to fit the "urban" and "country" classes.

For evaluation, the authors employ a range of metrics, including In Distribution Accuracy (I.D. Acc), BG Gap, CoObj Gap, and BG+CoObj Gap. These metrics gauge both the model's overall performance and its robustness in handling group shifts due to individual or combined spurious attributes.

### E.2 Hyperparameters

We used SGD as the optimization algorithm to maintain consistency with the existing literature. The hyperparameters employed in our experiments on spurious benchmarks are detailed in Tab. 5. For the Waterbirds, CelebA and UrbanCars datasets, we tuned the learning rate within the range of {1e-4, 1e-5} and weight decay within the range of {1e-1, 1e-0}. These ranges were determined based on the ranges of optimal hyperparameters used by the current state-of-the-art algorithms (Creager et al., 2021; Liu et al., 2021; Sagawa et al., 2019; Nam et al., 2021; Zhang et al., 2022). The batch sizes and total training epochs remained consistent with those used in these prior studies. To determine the epoch for separating groups, we performed clustering on the validation set while training the model on the training set to maximize the minimum recall of Spare's clusters with the groups in the validation set. As mentioned in Sec. 5, we decided the number of clusters and adjusted the sampling power for each class based on Silhouette scores. Specifically, when the Silhouette score was below 0.9, a sampling power of 2 or 3 was applied, while a sampling power of 1 was used otherwise. It is important to note that other algorithms tuned hyperparameters, such as epochs to separate groups and upweighting factors, by maximizing the worst-group accuracy of fully trained models on the validation set, which is more computationally demanding than the hyperparameter tuning of Spare.

Table 5: Hyperparameters used for the reported results on different datasets.

| Dataset | CMNIST | Waterbirds | CelebA | UrbanCars |
|---|---|---|---|---|
| Learning rate | 1e-3 | 1e-4 | 1e-5 | 1e-4 |
| Weight decay | 1e-3 | 1e-1 | 1e-0 | 1e-1 |
| Batch size | 32 | 128 | 128 | 128 |
| Training epochs | 20 | 300 | 50 | 300 |
| Group separation epoch | 2 | 2 | 1 | 2 |
| Silhouette scores | [0.997,0.978,0.996,0.991,0.996] | [0.886,0.758] | [0.924,0.757] | [0.849,0.872] |
| Sampling power | [1,1,1,1,1] | [3,3] | [1,2] | [2,2] |

### E.3 Choices of Model Outputs

In our experiments, we found the worst-group accuracy gets the most improvement when Spare uses the outputs of the last linear layer to separate the majority from the minority for CMNIST, Waterbirds and UrbanCars and use the second to last layer (i.e., the feature embeddings inputted to the last linear layer) to identify groups in CelebA. We speculate that this phenomenon can be attributed to the increased complexity of the CelebA dataset compared to the other two datasets, as employing a higher output dimension help identify groups more effectively.

### E.4 Dependency on the Clustering Algorithm

The performance of Spare is not sensitive to the clustering algorithm. The key to Spare is **clustering the entire model output early in training**. While $k$-means easily scales to medium-sized datasets, $k$-median is

Table 6: Wall-clock times for k-means clustering on Waterbirds, CelebA, CMNIST, and Restricted ImageNet datasets.

| **CMNIST** | **Waterbirds** | **Celeba** | **UrbanCars** | **Restricted ImageNet** |
|---:|---:|---:|---:|---:|
| 0.46s | 0.07s | 31.8s | 0.57s | 2s |

Table 7: Wall-clock runtime comparison of Spare and SOTA 2-stage algorithms.

| **ERM** | **JTT** | **CnC** | **SSA** | **Spare** |
|---|---|---|---|---|
| 1h12m | 9h5m | 4h25m | 2h15m | 1h16m |

more suitable for very large datasets, as it can be formulated as a submodular maximization problem (Wolsey, 1982) for which fast and scalable distributed (Mirzasoleiman et al., 2013, 2015) and streaming (Badanidiyuru et al., 2014) algorithms are available.

### E.5 Clustering Details

Clustering was performed on all data samples within the same class. It's important to note that k-means doesn't require loading all the data into memory and operates in a streaming manner. As an alternative, we also discussed the possibility of using the k-medoids clustering algorithm and its distributed implementation which uses submodular optimization and easily scales to millions of examples in Sec. 5. In Tab. 6, we present the wall-clock times for k-means clustering on CMNIST, Waterbirds, CelebA, UrbanCars and Restricted ImageNet. It shows that the cost of clustering is negligible when compared to the cost of training.

### E.6 Training Cost

Tab. 7 shows a all-clock runtime comparison of Spare and SOTA 2-stage algorithms on Waterbirds. JTT initially trains the identification model for a specific number of epochs and then upsamples misclassified examples by a substantial factor to train the robust model. As a result, the training cost is influenced not just by the training of the identification model but also by the considerable volume of upsampled training data used in the robust model's training. For instance, in the case of CelebA, JTT trains the identification model for just one epoch but then upsamples all misclassified examples (approximately 1/10 of the training set) by a factor of 50. This leads to a training set roughly six times the original size. In this scenario, the large volume of upsampled training data significantly increases the training cost, while the training time for the identification model is almost negligible.

## F GRADCAM VISUALIZATIONS: Spare HELPS THE LEARNING OF CORE FEATURES.

Fig. 10 compares GradCAM (Selvaraju et al., 2017) visualizations depicting saliency maps for samples from Waterbirds with water and land backgrounds (left), and from CelebA with different genders (right), when ResNet50 is trained by ERM vs. Spare. Warmer colors indicate the pixels that the model considered more important for making the final classification, based on gradient activations. We see that training with Spare allows the model to learn the core feature, instead of the spurious features.

## G DISCOVERING SPURIOUS FEATURES

### G.1 Restricted ImageNet

We use Restricted ImageNet proposed in Tsipras et al. (2019) which contains 9 superclasses of ImageNet. The classes and the corresponding ImageNet class ranges are shown in Tab. 8.

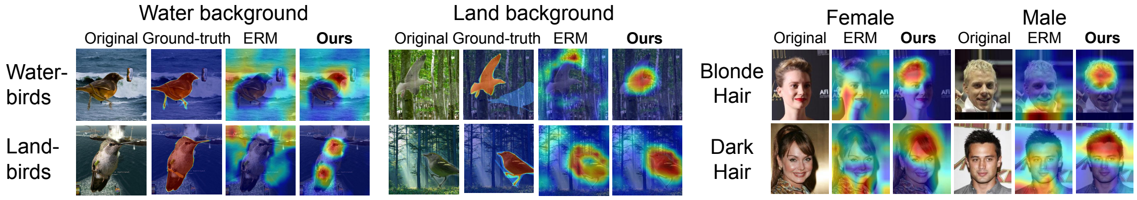


Figure 10: GradCAM Visualization. Warmer colors correspond to the pixels that are weighed more in making the final classification. Spare allows learning the core features instead of spurious ones.

Table 8: Classes included in Restricted ImageNet and their corresponding ImageNet class ranges.

| Restricted ImageNet Class | ImageNet class range |
|---|---|
| dog | 151-268 |
| cat | 281-285 |
| frog | 30-32 |
| turtle | 33-37 |
| bird | 80-100 |
| primate | 365-382 |
| fish | 389-397 |
| crab | 118-121 |
| insect | 300-319 |

### G.2 Experimental Settings

When training on Restricted ImageNet, we use ResNet50 (He et al., 2016) from the PyTorch library (Paszke et al., 2019) with randomly initialized weights instead of pretrained weights. We followed the hyperparameters specified in Goyal et al. (2017): the model was trained for 90 epochs, with an initial learning rate of 0.1. The learning rate was reduced by a factor of 0.1 at the 30th, 60th, and 80th epochs. During training, we employed Nesterov momentum of 0.9 and applied a weight decay of 0.0001.

### G.3 Investigation on Groups Identified by EIIL vs. Spare

**Evaluation Setup.** As no group-labeled validation set is available to tune the epoch in which the groups are separated, we tried separating groups using ERM models trained for various numbers of epochs. Since both EIIL and Spare identify the groups early (EIIL infers groups on models trained with ERM for 1 epoch for both Waterbirds and CelebA, and 5 epochs for CMNIST; the group separation epochs for Spare are epoch 1 or 2 for the three datasets, as shown in Tab. 5), we tuned the epoch to separate groups in the range of {2,4,6,8} for both algorithms. This tuning was based on the average test accuracy achieved by the final model, as the worst-group accuracy is undefined without group labels. Interestingly, while Spare did not show sensitivity to the initial epochs on Restricted ImageNet, EIIL achieved the highest average test accuracy when the initial models were trained for 4 epochs using ERM. We manually labeled examples with their groups for test data.

**EIIL finds groups of misclassified examples while Spare finds groups with spurious features.** We observed that EIIL effectively separates examples that have 0% classification accuracy as the minority group, as demonstrated in Tab. 9. This separation is analogous to the error-splitting strategy employed by JTT (Liu et al., 2021) when applied to the same initial model. This similarity in behavior is also discussed in (Creager et al., 2021). Instead of focusing on misclassified examples, Spare separates the examples that are learned early in training. Tab. 10 shows that the first cluster found by Spare have almost 100% accuracy, indicating that the spurious feature is learned for such examples. Downweighting examples that are learned early allows for effectively mitigating the spurious correlation.

**Spare upweights outliers less than EIIL.** Heavily upweighting misclassified examples can be problematic for this more realistic dataset than the spurious benchmarks as the misclassified ones are likely to be outliers, noisy-labeled or contain non-generalizable information. Tab. 9 shows that groups inferred by EIIL are more

Table 9: Accuracy (%) of training examples in different classes of Restricted ImageNet in the two environments inferred by EIIL. EIIL trains models with Group DRO on the inferred environments, resulting in up-weighting misclassified examples in Env 2.

| Class | dog | cat | frog | turtle | bird | primate | fish | crab | insect |
|---|---|---|---|---|---|---|---|---|---|
| Env 1 ERM acc | 98 | 37 | 26 | 62 | 76 | 78 | 78 | 71 | 90 |
| Env 2 ERM acc | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Env 1 size | 144378 | 488 | 457 | 2875 | 17157 | 11233 | 6817 | 2172 | 21112 |
| Env 2 size | 3495 | 6012 | 3443 | 3625 | 9984 | 12167 | 4417 | 3028 | 4888 |

imbalanced, which makes EIIL upweights misclassified examples more than Spare. As shown in Tab. 4, this heavier upweighting of misclassified examples with EIIL drops accuracy not only for the minority groups but also for the overall accuracy. Therefore, we anticipate that this effect would persist or become even more pronounced for methods like JTT, which directly identify misclassified examples as the minority group. In contrast, Spare separates groups based on the spurious feature that is learned early, and upweights the misclassified examples less than other methods due to the more balanced size of the clusters. This allows Spare to more effectively mitigate spurious correlations than others.

Table 10: Accuracy (%) of training examples in different classes of Restricted ImageNet in the two groups inferred by Spare at epoch 8.

| Class | dog | cat | frog | turtle | bird | primate | fish | crab | insect |
|---|---|---|---|---|---|---|---|---|---|
| Cluster 1 ERM acc | 100 | 100 | 100 | 100 | 100 | 99 | 100 | 100 | 100 |
| Cluster 2 ERM acc | 64 | 9 | 11 | 14 | 28 | 13 | 27 | 16 | 36 |
| Cluster 1 size | 130541 | 3236 | 1578 | 2684 | 18870 | 12158 | 7331 | 2566 | 18974 |
| Cluster 2 size | 17332 | 3264 | 2322 | 3816 | 8271 | 11242 | 3903 | 2634 | 7026 |

## H REPRODUCIBILITY

Each experiment was conducted on one of the following GPUs: NVIDIA A40 with 45G memory, NVIDIA RTX A6000 with 48G memory, and NVIDIA RTX A5000 with 24G memory.