Gradient Extrapolation for Debiased Representation Learning

Ihab Asaad[1] Maha Shadaydeh[1] Joachim Denzler[1]
[1]Computer Vision Group, Friedrich Schiller University Jena, Germany
{ihab.asaad, maha.shadaydeh, joachim.denzler}@uni-jena.de

Abstract

Machine learning classification models trained with empirical risk minimization (ERM) often inadvertently rely on spurious correlations. When absent in the test data, these unintended associations between non-target attributes and target labels lead to poor generalization. This paper addresses this problem from a model optimization perspective and proposes a novel method, Gradient Extrapolation for Debiased Representation Learning (GERNE), designed to learn debiased representations in both known and unknown attribute training cases. GERNE uses two distinct batches with different amounts of spurious correlations and defines the target gradient as a linear extrapolation of the gradients computed from each batch's loss. Our analysis shows that when the extrapolated gradient points toward the batch gradient with fewer spurious correlations, it effectively guides training toward learning a debiased model. GERNE serves as a general framework for debiasing, encompassing methods such as ERM, reweighting, and resampling, as special cases. We derive the theoretical upper and lower bounds of the extrapolation factor employed by GERNE. By tuning this factor, GERNE can adapt to maximize either Group-Balanced Accuracy (GBA) or Worst-Group Accuracy (WGA). We validate the proposed approach on five vision and one NLP benchmarks, demonstrating competitive and often superior performance compared to state-of-the-art baselines. The code is available at: https://gerne-debias.github.io/

1. Introduction

Deep learning models have demonstrated significant success in various classification tasks, but their performance is often compromised by datasets containing prevalent spurious correlations in the majority of samples [13, 18, 29, 52]. Spurious correlations refer to unintended associations between easy-to-learn non-target attributes and target labels, leading models based on Empirical Risk Minimization (ERM)- a widely used approach in classification tasks [45]- to rely on these correlations instead of the true, intrinsic features of the classes [10, 12, 40]. This occurs because the ERM objective optimizes for the average performance [45], which results in poor generalization when these spurious features are absent. For instance, in the Waterbirds classification task [46], where the goal is to classify a bird as either a waterbird or a landbird, the majority of waterbirds are associated with water backgrounds. In contrast, the majority of landbirds are associated with land backgrounds. A model trained with ERM might learn to classify the birds based on the background-water for waterbirds and land for landbirds-rather than focusing on the birds' intrinsic characteristics. This reliance on the spurious feature allows the model to perform well on the majority training samples, where these correlations hold, but fails to generalize to test samples where these correlations are absent (e.g., waterbirds on land). Examples of Waterbirds images shown in Fig. 1a. Avoiding spurious correlations is crucial across various applications, including medical imaging [25, 37], finance [11], and climate modeling [17].

This pervasive challenge has spurred extensive research into strategies for mitigating the negative effect of spurious correlations, particularly under varying levels of spurious attribute information availability. The authors of [50] provide a comprehensive review of the methods and research directions aimed at addressing this issue. In an ideal scenario, where attribute information is available in both the training and validation sets, methods can leverage this information to counteract spurious correlations [16, 39, 48]. When attribute information is available only in the validation set, methods either incorporate this set into the training process [18, 32, 42] or restrict its use to model selection and hyperparameter tuning [27–29, 31, 35]. Despite these efforts, existing methods still struggle to fully avoid learning spurious correlations, especially when the number of samples without spurious correlations is very limited in the training dataset, leading to poor generalization on the test data where these correlations are absent.

In this paper, we adopt a different research approach, seeking to address the issue of spurious correlations from a model optimization perspective. We propose a novel method, Gradient Extrapolation for Debiased Represen-

tation Learning (GERNE), to improve generalization and learn debiased representations. The contributions of this paper can be summarized as follows:

- We propose GERNE, a novel and easy-to-implement debiasing method in classification tasks. The core idea is to sample two types of batches with varying amounts of spurious correlations and compute the two losses on these two batches. We linearly extrapolate the gradients of these two losses to obtain a target gradient. The target gradient, controlled by an extrapolation factor, is used to update the model's parameters.
- The proposed gradient extrapolation approach is presented theoretically as a general framework for debiasing with methods, such as ERM, reweighting, and resampling, being shown as special cases.
- The extrapolation factor's theoretical upper and lower bounds are derived to ensure convergence, and its impact on performance is experimentally discussed.
- We also establish a link between the extrapolation factor and both the Group-Balanced Accuracy (GBA) and Worst-Group Accuracy (WGA) metrics and generalize GERNE to the case with unknown attributes.
- We highlight that in a biased dataset, overpresenting the minority groups in the sampled batches (compared to the majority) might be beneficial and can lead to SOTA results.
- We validate our approach on six benchmarks spanning both vision and NLP tasks, demonstrating superior performance compared to state-of-the-art methods.

2. Related Work

Debiasing according to attributes annotations availability. Numerous studies have leveraged attribute annotations to mitigate spurious correlations and learning debiased representation [3, 39, 51, 53, 55]. For instance, Group DRO [39] optimizes model performance on the worst-case group by minimizing worst-group error during training. While these methods are effective, obtaining attribute annotations for each sample can be extremely time-consuming and labor-intensive. Consequently, recent works have explored approaches that rely on limited attribute-labeled data to reduce the dependency on full annotations [18, 32, 42]. For example, DFR [18] enhances robustness by using a small, group-balanced validation set with attribute labels to retrain the final layer of a pre-trained model. For cases where attribute information is only available for model selection and hyperparameter tuning [6, 16, 28, 31, 54], usually an initial model is used to separate samples based on the alignment between the label and spurious attributes. Samples for which this model incurs relatively low loss are considered "easy" examples, where we expect the alignment to hold and the samples to closely resemble the majority group. In contrast, samples with high loss are considered "hard" examples, and these samples tend to resemble the minority group [49]. This process effectively creates "easy" and "hard" pseudo-attributes within each class, allowing debiasing methods that traditionally rely on attribute information to be applied. For example, JTT [28] first trains a standard ERM model and then trains a second model by upweighting the misclassified training examples detected by the first model. Finally, a more realistic and challenging scenario arises when attribute information is entirely unavailable [4, 43]-not accessible for training, model selection, or hyperparameter tuning-requiring models to generalize without explicit guidance on non-causal features [50].

Debiasing via balancing techniques. A prominent family of solutions to mitigate spurious correlations across the aforementioned scenarios of annotation availability involves data balancing techniques [7, 16, 19, 21, 36, 40, 47]. These methods are valued for their simplicity and adaptability, as they are typically faster to train and do not require additional hyperparameters. Resampling underrepresented groups to ensure a more balanced distribution of samples [16, 19] or modifying the loss function to adjust for imbalances [38] are common examples of these techniques. We demonstrate in Sec. 5.3 that although the balancing techniques are effective, their performance is constrained in the presence of spurious correlations. In contrast, our proposed debiasing approach mitigates the negative effects of spurious correlations by guiding the learning process in a debiasing direction, proving to be more effective.

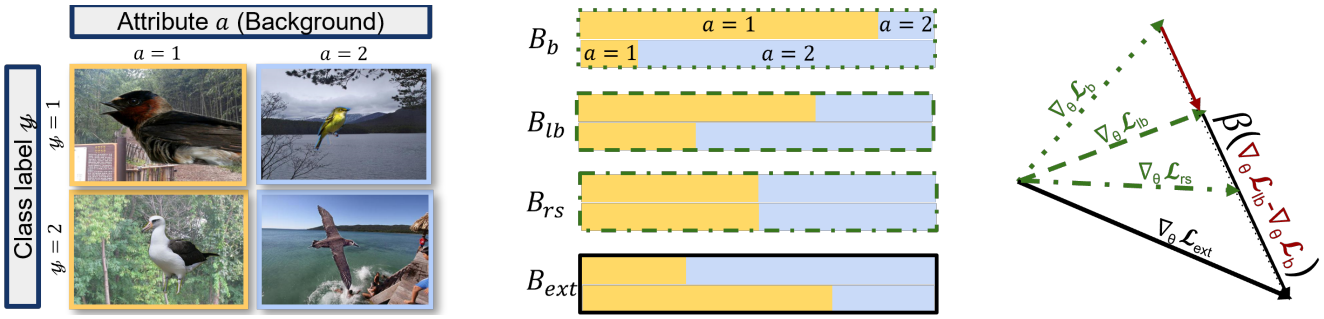
3. Problem Setup

We consider a standard multi-class classification problem with K classes and A spurious attributes. Each input sample $x_i \in \mathcal{X} = \{x_j \mid j = 1, \dots, N\}$ is associated with a class label $y_i \in \mathcal{Y} = \{1, \dots, K\}$ and an attribute $a_i \in \mathcal{A} = \{1, \dots, A\}$, where N is the total number of samples in the dataset. We define a group $\mathcal{X}_{y,a}$ for $(y, a) \in \mathcal{G} = \mathcal{Y} \times \mathcal{A}$ as the set of input samples x_i with class label y and attribute a, resulting in $|\mathcal{G}| = K \cdot A$ groups. For each class y, we denote by $\mathcal{X}_y = \bigcup_{a \in \mathcal{A}} \mathcal{X}_{y,a}$ the set of all samples with label y. We assume all groups are non-empty, i.e., $\forall (y, a) \in \mathcal{G}, \mathcal{X}_{y,a} \neq \emptyset$, and denote the cardinality of any group $\mathcal{X}_m$ by $|\mathcal{X}_m|$.

Our goal is to learn the intrinsic features that define the labels rather than spurious features present in a biased dataset. This would ensure robust generalization when spurious correlations are absent in the test distribution. Following [39], we aim to learn a function parameterized by a neural network $f^* : \mathcal{X} \to \mathbb{R}^K$ to minimize the risk for the worst-case group:

$$f^* = \arg\min_f \max_{g \in \mathcal{G}} \mathbb{E}_{x \sim p(x|(y,a)=g)} \left[\ell(y, f(x)) \right], \qquad (1)$$

where $\ell(y, f(x)) \to \mathbb{R}$ is the loss function.



(a) (b) (c)

Figure 1. (a): Sample images from the waterbirds classification task. Most landbird images appear with land backgrounds (i.e., $y = 1, a = 1$), while most waterbird images appear with water backgrounds (i.e., $y = 2, a = 2$). This correlation between bird class and background introduces spurious correlations in the dataset. (b): Visualization of batch construction. B_b shows a biased batch where the majority of images from class $y = 1$ (top row) have attribute $a = 1$ (yellow), and most images from class $y = 2$ (bottom row) have attribute $a = 2$ (light-blue). B_{lb} represents a less biased batch, with a more balanced attribute distribution within each class, controlled by c (here $c = \frac{1}{2}$). B_{rs} depicts a group-balanced distribution and refers to batch sampled using the Resampling method [16]. B_{ext} simulates GERNE's batch with $c \cdot (\beta + 1) > 1$, where the dataset's minority group appears as the majority in the batch. (c): A simplified 2D representation of gradient extrapolation where $\theta \in \mathbb{R}^2$. $\nabla_\theta \mathcal{L}_b$ is the gradient computed on B_b; training with this gradient is equivalent to training with ERM objective. $\nabla_\theta \mathcal{L}_{lb}$ represents the gradient computed on B_{lb}. $\nabla_\theta \mathcal{L}_{rs}$ is the gradient computed on B_{rs}, which is equivalent to an extraplated gradient with $c \cdot (\beta + 1) = 1$. Finally, $\nabla_\theta \mathcal{L}_{ext}$ is our extrapolated gradient, with the extrapolation factor β modulating the degree of debiasing in conjunction with the strength of spurious correlations present in the dataset.

4. The Proposed Method: GERNE

We build GERNE with the goal of mitigating the impact of spurious correlations. The core idea of GERNE is to sample two batches with different amounts of spurious correlations, hereafter named the biased batch $\boldsymbol{B_b}$ and the less biased batch $\boldsymbol{B_{lb}}$ (Fig. 1b). Let $\mathcal{L}_b, \mathcal{L}_{lb}$ be the losses calculated on $\boldsymbol{B_b}$ and $\boldsymbol{B_{lb}}$, respectively. We assume that extrapolating the gradients of these two losses towards the gradient of $\mathcal{L}_{lb}$ guides the model toward debiasing as illustrated in Fig. 1c. We first present GERNE for training with known attributes and then generalize GERNE to the unknown attribute case.

4.1. GERNE for the Known Attributes Case

In the following, we denote by $p(y, a)$ the joint distribution of class label y and attribute a in a sampled batch. During training, we construct two types of batches with different conditional attribute distributions $p(a|y)$: the *biased* and the *less biased* batches. Our method defines the target loss as a linear extrapolation between the losses computed on these two batches. A simplified illustration is shown in Fig. 1. Finally, we derive the link between the extrapolation factor and the risk of the worst-case group in Eq. (1), and theoretically define the upper and lower bounds of this factor.

4.1.1. Sampling the biased and the less biased batches

The biased batch and the less biased batches are sampled to satisfy the following two conditions:

1. Uniform sampling from classes, i.e., $\forall y \in \mathcal{Y}, p(y) = \frac{1}{K}$.
2. Uniform sampling from groups, i.e., $\forall (y, a) \in \mathcal{G}, p(x|y, a) = \frac{1}{|\mathcal{X}_{y,a}|}$ for $x \in \mathcal{X}_{y,a}$.

The **biased batch ($\boldsymbol{B_b}$)** is sampled with a conditional attribute distribution $p_b(a|y)$ within each class y to reflect the inherent bias present in the dataset. Specifically, $p_b(a|y) = \alpha_{ya}$, where:

$$\alpha_{ya} = \frac{|\mathcal{X}_{y,a}|}{|\mathcal{X}_y|}. \tag{2}$$

Note that to sample a biased batch, no access to the attributes is required, and uniformly sampling from $\mathcal{X}_y$ for each label y satisfies Eq. (2). The **less biased batch ($\boldsymbol{B_{lb}}$)** is sampled with a conditional attribute distribution, denoted as $p_{lb}(a|y)$, which satisfies the following: $\forall (y, a) \in \mathcal{G}$:

$$\min(\frac{1}{A}, p_b(a|y)) \leq p_{lb}(a|y) \leq \max(\frac{1}{A}, p_b(a|y)). \tag{3}$$

That is, B_{lb} exhibits a more balanced group distribution than B_b, and $\mathcal{L}_{lb}$ quantifies the loss when spurious correlations are reduced in the sampled batch. Choosing

$$p_{lb}(a|y) = (1-c) \cdot p_b(a|y) + c \cdot \frac{1}{A} = \alpha_{ya} + c \cdot (\frac{1}{A} - \alpha_{ya}) \tag{4}$$

satisfies the inequality in Eq. (3), where $c \in (0, 1]$ is a hyperparameter that controls the degree of bias reduction. An example of the two types of batches is presented in Fig. 1b.

4.1.2. Gradient extrapolation

We define our target loss $\mathcal{L}_{ext}$ as follows:

$$\mathcal{L}_{ext} = \mathcal{L}_{lb} + \beta \cdot (\mathcal{L}_{lb} - \mathcal{L}_b), \tag{5}$$

where β is a hyperparameter, and the loss form given the joint distribution $p(x, y, a)$ is defined as:

$$\mathcal{L} = \mathbb{E}_{(x,y,a)\sim p(x,y,a)} \left[\ell(y, f(x))\right]. \tag{6}$$

Given the set of parameters θ of our model f, the gradient of $\mathcal{L}_{ext}$ with respect to θ can be derived from Eq. (5):

$$\nabla_\theta \mathcal{L}_{ext} = \nabla_\theta \mathcal{L}_{lb} + \beta \cdot (\nabla_\theta \mathcal{L}_{lb} - \nabla_\theta \mathcal{L}_b). \tag{7}$$

Our target gradient vector $\nabla_\theta \mathcal{L}_{ext}$ in Eq. (7) is a linear extrapolation of the two gradient vectors $\nabla_\theta \mathcal{L}_{lb}$ and $\nabla_\theta \mathcal{L}_b$, and accordingly, we refer to β as the extrapolation factor. Because the less biased batch has a less skewed conditional attribute distribution compared to the biased batch (as shown in Eq. (3)), extrapolating their gradients and toward the less biased gradient forms a new gradient ($\mathcal{L}_{ext}$) that leads to learning even more debiased representation for some values of the extrapolation factor $\beta > 0$. A visual representation of extrapolation is shown in Fig. 1c.

4.1.3. GERNE as general framework for debiasing

Minimizing our target loss $\mathcal{L}_{ext}$ simulates minimizing the loss of class-balanced batches with the following conditional distribution of $(y, a) \in \mathcal{G}$:

$$p_{ext}(a|y) = \alpha_{ya} + c \cdot (\beta + 1) \cdot \left(\frac{1}{A} - \alpha_{ya}\right). \tag{8}$$

We provide the full proof in Appendix A.

Based on Eq. (8), we can establish the link between GERNE and other methods for different values of β, c:

- For $\beta = -1$, $\mathcal{L}_{ext} = \mathcal{L}_b$ and GERNE is equivalent to class-balanced ERM method.
- For $c = 1$ and $\beta = 0$, $p_{ext}(a|y) = \frac{1}{A}$, and GERNE matches the resampling method [16], which samples equally from all groups (B_{rs} in Fig. 1b, with gradient of the loss computed on it denoted as $\nabla_\theta \mathcal{L}_{rs}$ in Fig. 1c).
- For $c \cdot (\beta + 1) = 1$, we also have $p_{ext}(a|y) = \frac{1}{A}$, and $\mathcal{L}_{ext}$ is, in expectation, equivalent to $\mathcal{L}_{rs}$. However, their loss variances differ. In fact, GERNE permits controlling the variance of its loss through its hyperparameters (c, β), which may help escape sharp minima [1] and improve generalization [23]. The derivation of the variance of GERNE's loss is detailed in Appendix B.
- For $c \cdot (\beta + 1) > 1$, $p_{ext}(a|y) > \frac{1}{A}$ if $\alpha_{ya} < \frac{1}{A}$ (also $p_{ext}(a|y) < \frac{1}{A}$ if $\alpha_{ya} > \frac{1}{A}$). In this case, GERNE simulates batches where the underrepresented groups (i.e., those with $\alpha_{ya} < \frac{1}{A}$) are oversampled.

4.1.4. Upper and lower bounds of β

Having $p_{ext}(a|y)$ in Eq. (8) within $[0, 1]$, β should satisfy:

$$\max_{\substack{(y,a)\in\mathcal{G} \\ \alpha_{ya}\neq\frac{1}{A}}} \min(i^1_{ya}, i^2_{ya}) \leq \beta \leq \min_{\substack{(y,a)\in\mathcal{G} \\ \alpha_{ya}\neq\frac{1}{A}}} \max(i^1_{ya}, i^2_{ya}), \tag{9}$$

where: $i^1_{ya} = -\frac{\alpha_{ya}}{c\cdot(\frac{1}{A}-\alpha_{ya})} - 1, i^2_{ya} = \frac{1-\alpha_{ya}}{c\cdot(\frac{1}{A}-\alpha_{ya})} - 1$.
These bounds are used when tuning β. In Appendix C, we simplify these bounds to $[\beta_{\min}, \beta_{\max}] = [-1, i^1_{y''a''}]$, where $(y'', a'') = \arg\max_{(y,a)\in\mathcal{G}} \alpha_{ya}$. Note that β doesn't affect $p_{ext}(a|y)$ for $\alpha_{ya} = \frac{1}{A}$ according to Eq. (8).

4.1.5. Tuning β to minimize the risk of worst-case group

Eq. (5) can be rewritten as follows (detailed in Appendix A):

$$\mathcal{L}_{ext} = \frac{1}{K} \cdot \sum_{g=(y,a)\in\mathcal{G}} p_{ext}(a|y)(\beta) \cdot L_g, \tag{10}$$

where

$$L_g = \mathbb{E}_{x\sim p(x|(y,a)=g)} \left[\ell(y, f(x))\right]. \tag{11}$$

In the presence of spurious correlations, minority or less-represented groups often experience higher risks, primarily due to the model's limited exposure to these groups during training [20]. Taking this into consideration, we define $g' = (y', a') = \arg\min_{(y,a)\in\mathcal{G}} \alpha_{ya}$. Since $L_{g'}$ is weighted by $p_{ext}(a'|y')$, increasing β beyond $\frac{1}{c} - 1$ assigns more weight to $L_{g'}$ in Eq. (10) than any other group loss (all groups' losses are equally weighted when $c \cdot (\beta + 1) = 1$). This increase in β encourages the model to prioritize reducing the loss of the underrepresented group g' during training, therefore minimizing the risk of the worst-case group.

We outline the detailed steps of our approach for the known attribute case in Algorithm 1.

Algorithm 1 GERNE for the known attribute case

Input: $\mathcal{X}_{y,a} \subseteq \mathcal{X}$ for $y \in \mathcal{Y}$ and $a \in \mathcal{A}$, f with initial $\theta = \theta_0$, # epochs E, batch size per label B, # classes K, # attributes A, learning rate η.

Choose $c \in (0, 1]$ and $\beta \in [\beta_{\min}, \beta_{\max}]$ via grid search.
for epoch $= 1$ to E **do**
  Biased Batch $B_b = \emptyset$, Less Biased Batch $B_{lb} = \emptyset$
  for $(y, a) \in \mathcal{G}$ **do**
    Sample a mini-batch $B^{y,a}_b = \{(x, y)\} \subseteq \mathcal{X}_{y,a}$ of size $\alpha_{y,a} \cdot B$;
    $B_b = B_b \cup B^{y,a}_b$
    Sample a mini-batch $B^{y,a}_{lb} = \{(x, y)\} \subseteq \mathcal{X}_{y,a}$ of size $((1 - c) \cdot \alpha_{y,a} + \frac{c}{A}) \cdot B$
    $B_{lb} = B_{lb} \cup B^{y,a}_{lb}$
  end for
  Compute $\mathcal{L}_b, \mathcal{L}_{lb}$ on B_b, B_{lb}, respectively. Then, compute $\nabla_\theta \mathcal{L}_b$ and $\nabla_\theta \mathcal{L}_{lb}$.
  Compute $\nabla_\theta \mathcal{L}_{ext}$ using Eq. (7).
  Update parameters (SGD): $\theta \leftarrow \theta - \eta \cdot \nabla_\theta \mathcal{L}_{ext}$
end for

4.2. GERNE for the Unknown Attributes Case

If the attributes are unavailable during training, it is not possible to directly sample less biased batches. To address this,

we follow the previous work [28, 31, 54] by training a standard ERM model $\tilde{f}$ and using its predictions to create pseudo-attributes $\tilde{a}$. Since $\tilde{f}$ is trained on biased batches, it tends to rely on spurious correlations, resulting in biased predictions. Leveraging these predictions, we classify samples into easy—those with high-confidence predictions, where the spurious correlations likely hold— and hard — those with low-confidence predictions, where the spurious correlations may not hold. After training $\tilde{f}$, we select a threshold $t \in (0, 1)$ and construct pseudo-attributes based on model predictions as follows: For each class y, we compute the predictions $\tilde{y}_i = p(y|x_i) = \text{softmax}(\tilde{f}(x_i))_y$ for each $x_i \in \mathcal{X}_y$. We then split them into two non-empty subsets: The first subset contains the smallest $\lfloor t \cdot |\mathcal{X}_y| \rfloor$ values, and the corresponding samples form the group $\mathcal{X}_{y,\tilde{a}=1}$. The remaining samples forms the group $\mathcal{X}_{y,\tilde{a}=2}$. This process ensures that each set $\mathcal{X}_y$ is divided into two disjoint and non-empty groups. Consequently, the pseudo-attribute space consists of two values, denoted as $\tilde{\mathcal{A}} = \{1, 2\}$ (i.e., $\tilde{A} = 2$) with $\tilde{\mathcal{G}} = \mathcal{Y} \times \tilde{\mathcal{A}}$ replacing $\mathcal{G}$ in the unknown attribute case. t is a hyperparameter, and we outline the detailed steps of GERNE for the unknown case in Appendix D.

4.2.1. Tuning β to control the unknown conditional distribution of an attribute a in class y

After creating the pseudo-attributes and defining the pseudo-groups, we consider forming a new batch of size B by uniformly sampling $\gamma \cdot B$ examples from group $\mathcal{X}_{y,\tilde{a}=1}$ and $(1-\gamma) \cdot B$ examples from group $\mathcal{X}_{y,\tilde{a}=2}$, where $\gamma \in [0, 1], \gamma \cdot B \in \mathbb{N}$. The resulting conditional distribution of an attribute a given y in the constructed batch is:

$$p_B(a|y) = \sum_{\tilde{a} \in \tilde{\mathcal{A}}} p_B(\tilde{a}|y) \cdot p(a|\tilde{a}, y) \tag{12}$$

Because the max/min value of a linear program must occur at a vertex, we have for $p(a|\tilde{a}, y) = p_{\tilde{a},y}(a)$:

$$\forall \gamma \in [0, 1], \min_{\tilde{a} \in \tilde{\mathcal{A}}} p_{\tilde{a},y}(a) \leq p_B(a|y) \leq \max_{\tilde{a} \in \tilde{\mathcal{A}}} p_{\tilde{a},y}(a) \tag{13}$$

This means that if: $\max_{\tilde{a}} p_{\tilde{a},y}(a) < \frac{1}{A} (\min_{\tilde{a}} p_{\tilde{a},y}(a) > \frac{1}{A})$, then there is no value for γ can yield a batch with $p_B(a|y) > \frac{1}{A} (p_B(a|y) < \frac{1}{A})$ via sampling from the pseudo-groups.

Proposition 1. Creating a biased and less biased batch with the pseudo-attributes $\tilde{\mathcal{A}} = 2$, GERNE can simulate creating batches with more controllable conditional attribute distribution (i.e., $p_B(a|y) > \max_{\tilde{a} \in \tilde{\mathcal{A}}} p_{\tilde{a},y}(a)$ or $p_B(a|y) < \min_{\tilde{a} \in \tilde{\mathcal{A}}} p_{\tilde{a},y}(a)$). We provide the proof of this proposition in Appendix E.

5. Experiments

To assess the general applicability of GERNE, we evaluate its performance on five computer vision datasets and one natural language processing dataset: Colored MNIST (C-MNIST) [3, 27], Corrupted CIFAR-10 (C-CIFAR-10) [15, 31], Biased FFHQ (bFFHQ) [22, 27], Waterbird [46], CelebA [30], and CivilComments [5]. We categorize these datasets into two groups: Datasets-1 and Datasets-2. Datasets-1 includes the first three datasets mentioned above and is used to evaluate GERNE's performance without data augmentation. Datasets-2 includes the remaining three datasets, where our implementation follows the setup in [50] to ensure a fair comparison.

5.1. Experiments on Datasets-1

Datasets. The C-MNIST dataset represents an extension of the MNIST dataset [26], incorporating colored digits. Each digit is highly correlated with a specific color, which constitutes its majority group. In the C-CIFAR-10 dataset, each category of images is corrupted with a specific type of texture noise [15]. The bFFHQ dataset comprises human face images, with "age" and "gender" as the target and spurious attributes, respectively. The majority of images depicting females are labeled as "young" while the majority of images depicting males are labeled as "old".

Evaluation metrics. For both C-MNIST and C-CIFAR-10, we train with varying ratios of the minority to majority examples (0.5%, 1%, 2%, and 5%), and we follow the evaluation setup of [29, 31] using GBA on the test set as our evaluation metric. For bFFHQ, we train models with a 0.5% minority ratio and evaluate the performance based on the accuracy of the minority group in line with [27].

Baselines. We consider six baseline methods: For the known attribute case, we compare GERNE with Group DRO [39]. For the unknown attribute case, our baselines are ERM [45], JTT [28], LfF [31], DFA [27], LC [29], and DeNetDM [44].

Implementation details. We adopt the same model architectures as the baseline methods and utilize the SGD optimizer with a momentum of 0.9 and weight decay of 0.01 across all three datasets. Additional implementation details in Appendix F.1.

Results. Tab. 1 compares our results with baseline methods for both known and unknown attribute cases. The results of all baseline methods are adopted from [29] except for DeNetDM, which is sourced from [44]. When the attributes are known, GERNE outperforms Group DRO by a significant margin on C-MNIST and C-CIFAR-10 datasets. The improvement in performance ranges from about 5% on C-CIFAR-10 with 5% of minority group and up to 16% on C-MNIST with 1% of minority group. Furthermore,

Table 1. Performance comparison of GERNE and baseline methods on the C-MNIST, C-CIFAR-10, and bFFHQ datasets. We report the GBA (%) and standard deviation over three trials on the test set for C-MNIST and C-CIFAR-10, with varying ratios (%) of minority samples. For bFFHQ, we report the minority group accuracy (%). Baseline results are sourced from [29] as the same experimental settings are adopted. ✓/× indicate the presence/absence of attribute information in the training set, respectively. The best results are marked in bold, and the second-best are underlined.

Methods	Group Info	C-MNIST				C-CIFAR-10				bFFHQ
		0.5	1	2	5	0.5	1	2	5	
Group DRO	✓	63.12	68.78	76.30	84.20	33.44	38.30	45.81	57.32	-
GERNE ($c = 1, \beta = 0$)	✓	77.68 ±0.89	84.36 ±0.21	88.15 ±0.11	91.98 ±0.08	45.10 ±0.60	50.08 ±0.42	54.85 ±0.30	62.16 ±0.05	72.13 ±0.90
GERNE (ours)	✓	**77.79 ±0.90**	**84.47 ±0.37**	**88.30 ±0.20**	**92.16 ±0.10**	**45.34 ±0.60**	**50.84 ±0.17**	**55.51 ±0.10**	**62.40 ±0.27**	**85.20 ±0.86**
ERM	×	35.19 ±3.49	52.09 ± 2.88	65.86 ± 3.59	82.17 ± 0.74	23.08 ± 1.25	28.52 ± 0.33	30.06 ± 0.71	39.42 ± 0.64	56.70 ± 2.70
JTT	×	53.03 ± 3.89	62.90 ± 3.01	74.23 ± 3.21	84.03 ± 1.10	24.73 ± 0.60	26.90 ± 0.31	33.40 ± 1.06	42.20 ± 0.31	65.30 ± 2.50
LfF	×	52.50 ± 2.43	61.89 ± 4.97	71.03 ± 1.14	84.79 ± 1.09	28.57 ± 1.30	33.07 ± 0.77	39.91 ± 1.30	50.27 ± 1.56	62.20 ± 1.60
DFA	×	65.22 ± 4.41	81.73 ± 2.34	84.79 ± 0.95	89.66 ± 1.09	29.75 ± 0.71	36.49 ± 1.79	41.78 ± 2.29	51.13 ± 1.28	63.90 ± 0.30
LC	×	71.25 ± 3.17	82.25 ± 2.11	86.21 ± 1.02	91.16 ± 0.97	34.56 ± 0.69	37.34 ± 1.26	47.81 ± 2.00	54.55 ± 1.26	69.67 ± 1.40
DeNetDM	×	74.72 ± 0.99	**85.22 ± 0.76**	**89.29 ± 0.51**	**93.54 ± 0.22**	38.93 ± 1.16	44.20 ± 0.77	47.35 ± 0.70	56.30 ± 0.42	75.70 ± 2.80
GERNE (ours)	×	**77.25 ± 0.17**	83.98 ± 0.26	87.41 ± 0.31	90.98 ± 0.13	**39.90± 0.48**	**45.60± 0.23**	**50.19± 0.18**	**56.53 ± 0.32**	**76.80 ± 1.21**

we show that we outperform the resampling method [16] ($c = 1, \beta = 0$) for all ratios. More discussion added to Appendix H. For bFFHQ, GERNE results in an improvement of over 13% in comparison with the resampling method. For the unknown attribute case, GERNE outperforms all baselines, except for C-MNIST with 1%, 2% (ranks second) and 5% of minority group, while maintaining a lower standard deviation, despite not employing any data augmentation techniques. At this 5% ratio, LC achieves slightly higher accuracy-likely benefiting from its use of data augmentation to enrich bias-conflicting samples. Though DeNetDM excels on C-MNIST, its hypothesis-that shallower networks isolate core attributes while deeper ones capture spurious features-relies on a simple bias structure and linear feature decodability tied to network depth. In contrast, GERNE demonstrates better generalization when complex spurious features are present.

5.2. Experiments on Datasets-2

Datasets. We evaluate GERNE on three commonly used datasets [50]: Waterbirds [46], CelebA [30] and CivilComments [5].

Evaluation metrics. We follow the same evaluation strategy from [50] for model selection and hyperparameter tuning. When attributes are known in both training and validation, we use the worst-group test accuracy as the evaluation metric. When attributes are unknown in training, but known in validation, we use the worst-group validation accuracy. When attributes are unavailable in both, we use the worst-class validation accuracy.

Baselines. For each dataset, we select the three best performing methods reported in [50]. We end up with ERM [45], Group DRO [39], DFR [18], LISA [51], ReSample [19], Mixup [53], ReWeightCRT [21], ReWeight [19], CBLoss [7], BSoftmax [36] and SqrtReWeight [50]. We also report the results for CnC [54] as it adopts similar training settings.

Implementation details. We employ the same data augmentation techniques, optimizers and pretrained models described in [50]. Further details are in Appendix F.2.

Results Tab. 2 shows the worst-group accuracy of the test set for GERNE against the baseline methods under the evaluation strategy explained above. In the case of known attributes, GERNE achieves the highest performance on the CelebA and CivilComments datasets and ranks second on Waterbirds, following DFR. In case of unknown attributes in training set but known in validation, our approach again achieves the best results on the Waterbirds and CivilComments datasets and remains competitive on CelebA, closely following the top two baseline results. Notably, DFR uses the validation set to train the model, whereas GERNE only uses it for model selection and hyperparameter tuning. We include a comparison of our method's performance against DFR, where GERNE also uses the validation set for training, in Appendix G. In the scenario where attributes are unknown in both the training and validation sets, our approach achieves the best results on the Waterbirds and CelebA datasets. However, we observe a significant drop in accuracy on CelebA compared to the second case (unknwon attributes in training but known in validation), while this drop is less pronounced on Waterbirds. This can be explained by the worst-class accuracy evaluation metric. In the validation set of CelebA, the majority examples in class 1 exhibit spurious correlations, leading to selection process to favor the majority group while disregarding the minority group. However, the validation set of Waterbirds has balanced groups within each class, resulting in only a slight performance drop for GERNE between the second and third

case. This highlights the critical role of having access to the attributes in the validation set or having a group-balanced validation set for model selection and hyperparameter tuning when aiming for better results with GERNE.

Table 2. Performance comparison of GERNE and baseline methods on the Waterbirds, CelebA, and CivilComments datasets. We report the worst-group test accuracy (%) and standard deviation over three trials on the test of each dataset. Baseline results are sourced from [50] as the same experimental settings are adopted. ✓/✓ indicates known attributes in training and validation sets. ×/✓ in validation set only, and ×/× in neither. Best results are highlighted in bold, and the second-best are underlined.

Methods	Group Info (train/val attr.)	Waterbirds	CelebA	Civil-Comments
ERM	✓/✓	69.10 ± 4.70	62.60 ± 1.50	63.70 ± 1.50
Group DRO	✓/✓	78.60 ± 1.00	89.00 ± 0.70	70.60 ± 1.20
ReWeight	✓/✓	86.90 ± 0.70	89.70 ± 0.20	65.30 ± 2.50
ReSample	✓/✓	77.70 ± 1.20	87.40 ± 0.80	73.30 ± 0.50
CBLoss	✓/✓	86.20 ± 0.30	89.40 ± 0.70	73.30 ± 0.20
DFR	✓/✓	**91.00 ± 0.30**	90.40 ± 0.10	69.60 ± 0.20
LISA	✓/✓	88.70 ± 0.60	86.50 ± 1.20	73.70 ± 0.30
GERNE (ours)	✓/✓	90.20 ± 0.22	**91.98 ± 0.15**	**74.65 ± 0.20**
ERM	×/✓	69.10 ± 4.70	57.60 ± 0.80	63.20 ± 1.20
Group DRO	×/✓	73.10 ± 0.40	78.50 ± 1.10	69.50 ± 0.70
ReWeight	×/✓	72.50 ± 0.30	81.50 ± 0.90	69.90 ± 0.60
DFR	×/✓	89.00 ± 0.20	86.30 ± 0.30	63.90 ± 0.30
Mixup	×/✓	78.20 ± 0.40	57.80 ± 0.80	66.10 ± 1.30
LISA	×/✓	78.20 ± 0.40	57.80 ± 0.80	66.10 ± 1.30
BSoftmax	×/✓	74.10 ± 0.90	83.30 ± 0.30	69.40 ± 1.20
ReSample	×/✓	70.00 ± 1.00	82.20 ± 1.20	68.20 ± 0.70
CnC	×/✓	88.50± 0.30	**88.80± 0.90**	68.90± 2.10
GERNE (ours)	×/✓	**90.21 ± 0.42**	86.28 ± 0.12	**71.00 ± 0.33**
ERM	×/×	69.10 ± 4.70	57.60 ± 0.80	63.20 ± 1.20
Group DRO	×/×	73.10 ± 0.40	68.30 ± 0.90	61.50 ± 1.80
DFR	×/×	89.00 ± 0.20	73.70 ± 0.80	64.40 ± 0.10
Mixup	×/×	77.50 ± 0.70	57.80 ± 0.80	65.80 ± 1.50
LISA	×/×	77.50 ± 0.70	57.80 ± 0.80	65.80 ± 1.50
ReSample	×/×	70.00 ± 1.00	74.10 ± 2.20	61.00 ± 0.60
ReWeightCRT	×/×	76.30 ± 0.20	70.70 ± 0.60	64.70 ± 0.20
SqrtReWeight	×/×	71.00 ± 1.40	66.90 ± 2.20	**68.60 ± 1.10**
CRT	×/×	76.30 ± 0.80	69.60 ± 0.70	67.80 ± 0.30
GERNE (ours)	×/×	**89.88 ± 0.67**	**74.24 ± 2.51**	63.10 ± 0.22

5.3. GERNE vs. Balancing Techniques

Balancing techniques have been shown to achieve state-of-the-art results, while remaining easy to implement [16, 50]. Notably, resampling methods often outperform reweighting strategies when combined with stochastic gradient algorithms [2]. Our results, as presented in Tab. 1, demonstrate that GERNE outperforms resampling (GERNE with $c = 1, \beta = 0$) when the evaluation metric is Group-Balanced Accuracy or Accuracy on minority group. More discussion in Appendix H. This highlights the flexibility of GERNE to adapt to maximize both metrics, and its superior performance in comparison to resampling and other balancing techniques, as further supported by the results in Tab. 2. In Appendix B, we provide a detailed ablation study comparing GERNE to an equivalent sampling+weighting approach with matching loss expectation, and demonstrate how GERNE leverages its controllable loss variance (by the hyperparameters c, β) to escape sharp minima.

5.4. Ablation Study

Tuning the extrapolation factor β. The value of β in Eq. (7) has a significant impact on the performance of our method in debiasing the model (i.e., leading the training process in a debiased direction and avoid learning spurious features). In Fig. 2, we show how tuning β affects the learning process in the case of the C-MNIST dataset with 0.5% of minority group in the known attribute case. We show results for $\beta \in \{-1, 0, 1, 1.2\}$ with $c = 0.5$. For $\beta = -1$, our target loss $\mathcal{L}_{ext}$ in Eq. (5) equals the biased loss $\mathcal{L}_b$, which leads to learning a biased model that exhibits high accuracy in the majority group, yet demonstrates poor performance on both the minority group and the unbiased test set. As β increases (e.g. $\beta = 0, \beta = 1$), the model starts learning more intrinsic features. This is evident from the improved performance on the minority group in the validation set, as well as on the unbiased test set. However, as the extrapolation factor β continues to increase, the model begins to exhibit higher variance during the training process as shown for $\beta = 1.2$, ultimately leading to divergence when β exceeds the upper bound defined in Eq. (9) which is equal to 1.22 in this case. While GERNE appears to be sensitive to small variations in β (e.g. 1.2 to 1.22), we show in Appendix C that β's upper bound is inversely proportional to c, A. By comparing the accuracies of Minority/Majority training groups in case $\beta = 0, \beta = 1$, we can see that both cases have around 100% accuracy on minority but higher accuarcy on majority for $\beta = 0$. However, we notice a better generalization when $\beta = 1$. This highlights the importance of directing the training process to the right direction early in training while overfitting is expected as well.

How the selection of t influences the optimal value of β. To answer this question, we conduct experiments on C-MNIST dataset with 0.5% of minority group. We first train a biased model $\tilde{f}$, and use its predictions to generate the pseudo-attributes for five different values of the threshold t. Let's refer to the pseudo-groups with $\tilde{a} = 1$ as the pseudo-minority groups. For each threshold, we tune β to achieve the best average accuracy on test set. Simultaneously, we compute the average precision and recall for the minority group. As shown in Fig. 3, with $t = 5 \times 10^{-4}$, the average precision reaches 1, indicating that all the samples in the pseudo-minority groups are from the minority group. However, these samples constitute less than 20% of the total number of samples in the minority group, as indicated by the average recall. Despite this, GERNE achieves a high accuracy of approximately 70%, remaining competitive with other methods reported in Tab. 1 while using only a very limited number of minority samples ($t = 5 \times 10^{-4}$

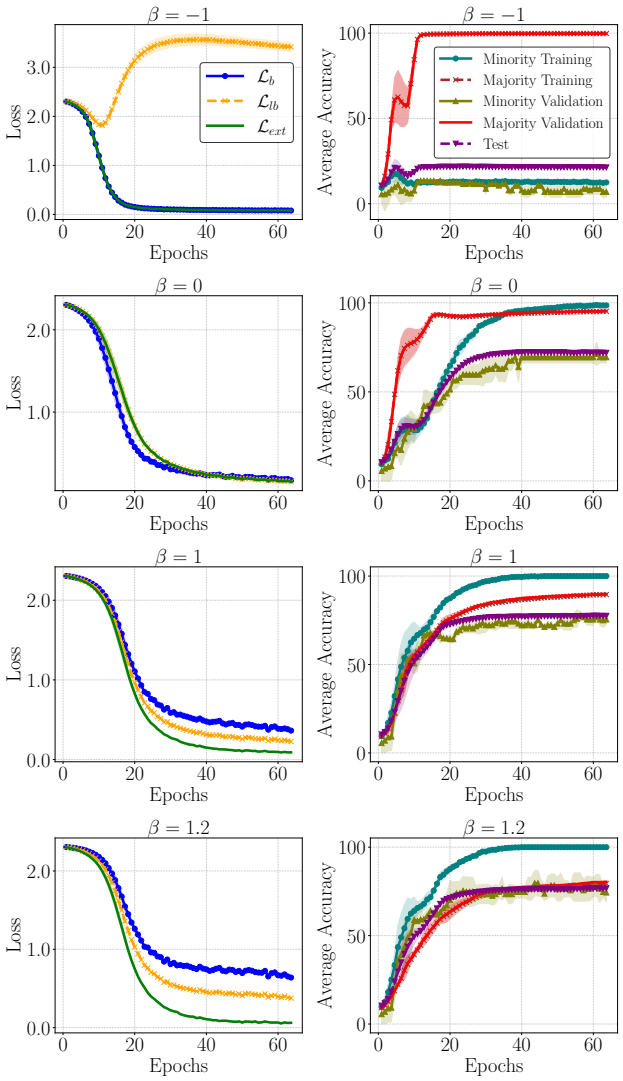


Figure 2. The impact of tuning β in debiasing the model $\beta \in \{-1, 0, 1, 1.2\}$ on debiasing the model. On the left column, we plot the training losses $\mathcal{L}_b, \mathcal{L}_{lb}$ and the target loss $\mathcal{L}_{ext}$. On the right column, we plot the average accuracy of the minority and majority groups in both training and validation, as well as the average accuracy of the unbiased test set. Each plot represents the mean and standard deviation calculated over three runs with different random seeds.

corresponds to about 28 samples versus 249 minority samples out of 55,000 samples in the training set). As t increases to 10^{-3} and 3×10^{-3}, precision remains close to 1 while increasing the number of minority samples in the pseudo-minority group. This increase introduces more diversity among minority samples within the pseudo-minority group, allowing for lower β values to achieve the best average accuracy on test set. However, for even higher thresholds, such as $t = 10^{-2}$, minority samples constitute less

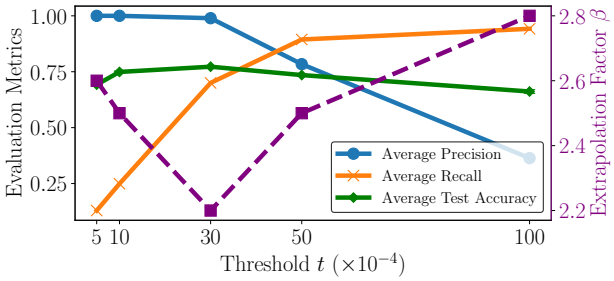


Figure 3. The effect of the threshold t used to generate pseudo-attributes on the extrapolation factor β and model performance. We plot the average precision and recall over pseudo-minority groups $(y, \tilde{a} = 1)$, averaged across all classes y. For each $(y, \tilde{a} = 1)$, precision is defined as the fraction of minority samples among all samples in that group, and recall is the fraction of those minority samples relative to all minority samples in class y. We also report the best achievable test accuracy, along with the corresponding extrapolation factor β, across different threshold values.

than 40% in the pseudo-minority group, prompting a need to revert to higher β values. We conclude that identifying the minority group is of utmost importance for achieving optimal results (high average precision and high recall) and this agrees with the results presented in both Tab. 1, Tab. 2 where we achieve the best results in the case of known attributes.

6. Conclusion

We introduce GERNE, a novel debiasing approach that effectively mitigates spurious correlations by leveraging an extrapolated gradient update. By defining a debiasing direction from loss gradients computed on batches with varying degrees of spurious correlations, GERNE's tunable extrapolation factor allows optimizing either Group-Balanced Accuracy (GBA) or Worst-Group Accuracy (WGA). Our comprehensive evaluations across vision and NLP benchmarks demonstrate GERNE's superior performance over state-of-the-art methods, both for known and unknown attribute cases, crucially achieving this without data augmentation. Furthermore, GERNE offers a unifying framework that encompasses methods like ERM and resampling, extending its applicability to unbiased datasets. Future work will explore dynamic adaptation of the extrapolation factor and refine attribute estimation for cases where attributes are unknown.

7. Acknowledgments

This work was funded by the Carl Zeiss Foundation within the project Sensorized Surgery, Germany (P2022-06-004). Maha Shadaydeh is supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Individual Research Grant SH 1682/1-1.

References

[1] Kwangjun Ahn, Ali Jadbabaie, and Suvrit Sra. How to escape sharp minima with random perturbations. In Proceedings of the 41st International Conference on Machine Learning, pages 597–618. PMLR, 2024. 4, 13

[2] Jing An, Lexing Ying, and Yuhua Zhu. Why resampling outperforms reweighting for correcting sampling bias with stochastic gradients. In International Conference on Learning Representations, 2021. 7

[3] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization, 2020. 2, 5, 14

[4] Saeid Asgari, Aliasghar Khani, Fereshte Khani, Ali Gholami, Linh Tran, Ali Mahdavi Amiri, and Ghassan Hamarneh. Masktune: Mitigating spurious correlations by forcing to explore. Advances in Neural Information Processing Systems, 35:23284–23296, 2022. 2

[5] Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. Nuanced metrics for measuring unintended bias with real data for text classification, 2019. 5, 6, 15

[6] Elliot Creager, Jörn-Henrik Jacobsen, and Richard Zemel. Environment inference for invariant learning. In International Conference on Machine Learning, pages 2189–2200. PMLR, 2021. 2

[7] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 9268–9277, 2019. 2, 6

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, pages 248–255. Ieee, 2009. 14

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers), pages 4171–4186, 2019. 15

[10] Mengnan Du, Fengxiang He, Na Zou, Dacheng Tao, and Xia Hu. Shortcut learning of large language models in natural language understanding. Communications of the ACM, 67 (1):110–120, 2023. 1

[11] Alessandro Fabris, Stefano Messina, Gianmaria Silvello, and Gian Antonio Susto. Algorithmic fairness datasets: the story so far. Data Mining and Knowledge Discovery, 36(6):2074–2152, 2022. 1

[12] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. Nature Machine Intelligence, 2(11):665–673, 2020. 1

[13] Tatsunori Hashimoto, Megha Srivastava, Hongseok Namkoong, and Percy Liang. Fairness without demographics in repeated loss minimization. In International Conference on Machine Learning, pages 1929–1938. PMLR, 2018. 1

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016. 14, 15

[15] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. arXiv preprint arXiv:1903.12261, 2019. 5, 14

[16] Badr Youbi Idrissi, Martin Arjovsky, Mohammad Pezeshki, and David Lopez-Paz. Simple data balancing achieves competitive worst-group-accuracy. In Conference on Causal Learning and Reasoning, pages 336–351. PMLR, 2022. 1, 2, 3, 4, 6, 7, 14, 15

[17] Fernando Iglesias-Suarez, Pierre Gentine, Breixo Solino-Fernandez, Tom Beucler, Michael Pritchard, Jakob Runge, and Veronika Eyring. Causally-informed deep learning to improve climate models and projections. Journal of Geophysical Research: Atmospheres, 129(4): e2023JD039202, 2024. 1

[18] Pavel Izmailov, Polina Kirichenko, Nate Gruver, and Andrew Gordon Wilson. On feature learning in the presence of spurious correlations, 2022. 1, 2, 6

[19] Nathalie Japkowicz. The class imbalance problem: Significance and strategies. In Proc. of the Int'l Conf. on artificial intelligence, pages 111–117, 2000. 2, 6

[20] Justin M Johnson and Taghi M Khoshgoftaar. The effects of data sampling with deep learning and highly imbalanced big data. Information Systems Frontiers, 22(5):1113–1131, 2020. 4

[21] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In International Conference on Learning Representations, 2020. 2, 6

[22] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 4401–4410, 2019. 5, 14

[23] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In International Conference on Learning Representations, 2017. 4, 13

[24] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Proceedings of the International Conference on Learning Representations (ICLR) 2015, 2015. 15

[25] Burak Koçak, Andrea Ponsiglione, Arnaldo Stanzione, Christian Bluethgen, João Santinha, Lorenzo Ugga, Merel Huisman, Michail E Klontzas, Roberto Cannella, and Renato Cuocolo. Bias in artificial intelligence for medical imaging: fundamentals, detection, avoidance, mitigation, challenges, ethics, and prospects. Diagnostic and Interventional Radiology, 31(2):75, 2025. 1

[26] Yann LeCun, Corinna Cortes, Chris Burges, et al. Mnist handwritten digit database, 2010. 5

[27] Jungsoo Lee, Eungyeup Kim, Juyoung Lee, Jihyeon Lee, and Jaegul Choo. Learning debiased representation via disentangled feature augmentation. In Advances in Neural Information Processing Systems, pages 25123–25133. Curran Associates, Inc., 2021. 1, 5, 14

[28] Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. In International Conference on Machine Learning, pages 6781–6792. PMLR, 2021. 2, 5

[29] Sheng Liu, Xu Zhang, Nitesh Sekhar, Yue Wu, Prateek Singhal, and Carlos Fernandez-Granda. Avoiding spurious correlations via logit correction. In The Eleventh International Conference on Learning Representations, 2023. 1, 5, 6

[30] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In 2015 IEEE International Conference on Computer Vision (ICCV), pages 3730–3738, 2015. 5, 6, 15

[31] Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. Learning from failure: De-biasing classifier from biased classifier. In Advances in Neural Information Processing Systems, pages 20673–20684. Curran Associates, Inc., 2020. 1, 2, 5, 14

[32] Junhyun Nam, Jaehyung Kim, Jaeho Lee, and Jinwoo Shin. Spread spurious attribute: Improving worst-group accuracy with spurious attribute estimation. arXiv preprint arXiv:2204.02070, 2022. 1, 2

[33] Arvind Neelakantan, Luke Vilnis, Quoc V. Le, Ilya Sutskever, Lukasz Kaiser, Karol Kurach, and James Martens. Adding gradient noise improves learning for very deep networks, 2015. 13

[34] Hyeonwoo Noh, Tackgeun You, Jonghwan Mun, and Bohyung Han. Regularizing deep neural networks by noise: Its interpretation and optimization. In Advances in Neural Information Processing Systems. Curran Associates, Inc., 2017. 13

[35] Shikai Qiu, Andres Potapczynski, Pavel Izmailov, and Andrew Gordon Wilson. Simple and fast group robustness by automatic feature reweighting. In International Conference on Machine Learning, pages 28448–28467. PMLR, 2023. 1

[36] Jiawei Ren, Cunjun Yu, shunan sheng, Xiao Ma, Haiyu Zhao, Shuai Yi, and hongsheng Li. Balanced meta-softmax for long-tailed visual recognition. In Advances in Neural Information Processing Systems, pages 4175–4186. Curran Associates, Inc., 2020. 2, 6

[37] María Agustina Ricci Lara, Rodrigo Echeveste, and Enzo Ferrante. Addressing fairness in artificial intelligence for medical imaging. nature communications, 13(1):4581, 2022. 1

[38] T-YLPG Ross and GKHP Dollár. Focal loss for dense object detection. In proceedings of the IEEE conference on computer vision and pattern recognition, pages 2980–2988, 2017. 2

[39] Shiori Sagawa*, Pang Wei Koh*, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks. In International Conference on Learning Representations, 2020. 1, 2, 5, 6

[40] Shiori Sagawa, Aditi Raghunathan, Pang Wei Koh, and Percy Liang. An investigation of why overparameterization exacerbates spurious correlations. In International Conference on Machine Learning, pages 8346–8356. PMLR, 2020. 1, 2

[41] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. Journal of big data, 6(1):1–48, 2019. 15

[42] Nimit Sharad Sohoni, Maziar Sanjabi, Nicolas Ballas, Aditya Grover, Shaoliang Nie, Hamed Firooz, and Christopher Re. BARACK: Partially supervised group robustness with guarantees. In ICML 2022: Workshop on Spurious Correlations, Invariance and Stability, 2022. 1, 2

[43] Christos Tsirigotis, Joao Monteiro, Pau Rodriguez, David Vazquez, and Aaron C Courville. Group robust classification without any group information. Advances in Neural Information Processing Systems, 36:56553–56575, 2023. 2

[44] Silpa Vadakkeeveetil Sreelatha, Adarsh Kappiyath, Abhra Chaudhuri, and Anjan Dutta. Denetdm: Debiasing by network depth modulation. Advances in Neural Information Processing Systems, 37:99488–99518, 2024. 5

[45] V.N. Vapnik. An overview of statistical learning theory. IEEE Transactions on Neural Networks, 10(5):988–999, 1999. 1, 5, 6

[46] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. Cub-200-2011, 2022. 1, 5, 6, 15

[47] Xinyi Wang, Michael Saxon, Jiachen Li, Hongyang Zhang, Kun Zhang, and William Yang Wang. Causal balancing for domain generalization. arXiv preprint arXiv:2206.05263, 2022. 2

[48] Shirley Wu, Mert Yuksekgonul, Linjun Zhang, and James Zou. Discover and cure: Concept-aware mitigation of spurious correlation. In International Conference on Machine Learning, pages 37765–37786. PMLR, 2023. 1

[49] Yu Yang, Besmira Nushi, Hamid Palangi, and Baharan Mirzasoleiman. Mitigating spurious correlations in multi-modal models during fine-tuning. In International Conference on Machine Learning, pages 39365–39379. PMLR, 2023. 2

[50] Yuzhe Yang, Haoran Zhang, Dina Katabi, and Marzyeh Ghassemi. Change is hard: A closer look at subpopulation shift. In International Conference on Machine Learning, 2023. 1, 2, 5, 6, 7, 15

[51] Huaxiu Yao, Yu Wang, Sai Li, Linjun Zhang, Weixin Liang, James Zou, and Chelsea Finn. Improving out-of-distribution robustness via selective augmentation, 2022. 2, 6

[52] Wenqian Ye, Guangtao Zheng, Xu Cao, Yunsheng Ma, and Aidong Zhang. Spurious correlations in machine learning: A survey. arXiv preprint arXiv:2402.12715, 2024. 1

[53] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In International Conference on Learning Representations, 2018. 2, 6

[54] Michael Zhang, Nimit S. Sohoni, Hongyang R. Zhang, Chelsea Finn, and Christopher Ré. Correct-n-contrast: A contrastive approach for improving robustness to spurious correlations, 2024. 2, 5, 6

[55] Chunting Zhou, Xuezhe Ma, Paul Michel, and Graham Neubig. Examining and combating spurious features under distribution shift. In International Conference on Machine Learning, pages 12857–12867. PMLR, 2021. 2

Gradient Extrapolation for Debiased Representation Learning

Supplementary Material

A. Conditional attribute distribution of the extrapolated loss: Proof of Eq. (8)

$\mathcal{L}$ in Eq. (6) can be written as:

$$\mathcal{L} = \int \ell(y, f(x))p(y)\, p(a|y)\, p(x|a, y)\, dx\, dy\, da.$$

Therefore, $\mathcal{L}_b, \mathcal{L}_{lb}$ can be written as:

$$\mathcal{L}_b = \int \ell(y, f(x))p(y)\, p_b(a|y)\, p(x|a, y)\, dx\, dy\, da. \quad (14)$$

$$\mathcal{L}_{lb} = \int \ell(y, f(x))p(y)\, p_{lb}(a|y)\, p(x|a, y)\, dx\, dy\, da. \quad (15)$$

Where $p_b(a|y), p_{lb}(a|y)$ defined in Eq. (2), Eq. (4), respectively, and x is uniformly sampled within each group (y, a) (i.e. $p(x|y, a) = \frac{1}{|\mathcal{X}_{y,a}|}$). Substituting Eq. (14), Eq. (15) in Eq. (5):

$$\begin{aligned}
\mathcal{L}_{ext} &= \int \ell(y, f(x))\, p(y) \Big(p_{lb}(a|y) \\
&\quad + \beta \cdot (p_{lb}(a|y) - p_b(a|y))\Big) \cdot p(x|a, y)\, dx\, dy\, da \\
&= \int \ell(y, f(x))\, p(y) \left(\alpha_{ya} + c \cdot (\beta + 1) \cdot (\frac{1}{A} - \alpha_{ya})\right) \cdot \\
&\quad p(x|a, y)\, dx\, dy\, da \\
&= \int \ell(y, f(x))\, p(y)\, p_{ext}(a|y) \cdot p(x|a, y)\, dx\, dy\, da.
\end{aligned}$$

Then:

$$p_{ext}(a|y) = \alpha_{ya} + c \cdot (\beta + 1) \cdot (\frac{1}{A} - \alpha_{ya}).$$

Furthermore, we can write $\mathcal{L}_{ext}$ as follows:

$$\begin{aligned}
\mathcal{L}_{\text{ext}} &= \mathbb{E}_{y\sim p(y)} \left[\mathbb{E}_{a\sim p_{\text{ext}}(a|y)} \left[\mathbb{E}_{x\sim p(x|a,y)} \left[\ell(y, f(x))\right]\right]\right] \\
&= \frac{1}{K} \sum_{g=(y,a)\in\mathcal{G}} p_{\text{ext}}(a|y; \beta) \cdot L_g
\end{aligned}$$

with $L_g = \mathbb{E}_{x\sim p(x|(y,a)=g)} \left[\ell(y, f(x))\right]$ by using the discrete expectations over y and $a|y$, with $p(y) = \frac{1}{K}$.

B. GERNE versus an equivalent sampling and weighting approach

We compare GERNE with an equivalent (in term of loss expectation) sampling+weighting method, which we refer to as 'SW'. For simplicity, we assume the following:

1. A binary classification task where the number of classes equals the number of attributes (i.e. $K = A = 2$).
2. The attributes are known, and the classes are balanced. (i.e. $|\mathcal{X}_{y=1}| = |\mathcal{X}_{y=2}|$).
3. The majority of samples which hold the spurious correlation in each class are aligned with the class label (i.e. $|\mathcal{X}_{y,a=y}| > |\mathcal{X}_{y,a\neq y}|$).
4. The dataset is highly biased. In other words, $\frac{|\mathcal{X}_{y,a\neq y}|}{|\mathcal{X}_{y,a=y}|} \ll 1$.
5. In a highly biased dataset, best performance is coupled with overpresenting the minority (conflicting samples according to assumption 1.) in early stages of training. Therefore, an overfitting on the minority is expected before the overfitting on the majority.

We refer to the expected loss of the majority (aligned) samples as $\mathcal{L}_A$, and the expected loss of the minority (conflicting) as $\mathcal{L}_C$. For GERNE, we sample two batches: biased and less biased batch, each of size B. From Eq. (5), $\mathcal{L}_{ext}$ can be written as:

$$\mathcal{L}_{ext} = (1 + \beta) \cdot \mathcal{L}_{lb} - \beta \cdot \mathcal{L}_b \quad (16)$$

Since the biased batch reflects the inherent bias present in the dataset, under the third assumption, we can approximate $\mathcal{L}_b$ by $\mathcal{L}_A$, neglecting the loss on the very few conflicting samples in the batch. Therefore, we have:

$$\mathcal{L}_b \approx \mathcal{L}_A \quad (17)$$

Following the third assumption and the conditional attribute distribution in Eq. (4), we can approximate the composition of the less biased batch as follows: a proportion of $(1 - \frac{c}{2})$ of the samples in the less biased batch are drawn from the aligned samples, while a proportion of $\frac{c}{2}$ of the samples from the minority group. this leads to the following approximation:

$$\mathcal{L}_{lb} \approx (1 - \frac{c}{2}) \cdot \mathcal{L}_A + \frac{c}{2} \cdot \mathcal{L}_C \quad (18)$$

Substituting Eq. (17), Eq. (18) into Eq. (16):

$$\mathcal{L}_{ext} \approx \frac{2 - c \cdot (1 + \beta)}{2} \cdot \mathcal{L}_A + \frac{c \cdot (1 + \beta)}{2} \cdot \mathcal{L}_C. \quad (19)$$

We consider the following 'SW' approach:

- Sampling step : we sample an 'SW' batch of size B similar to the less biased batch in GERNE. Where $(1 - \frac{c}{2})$ of the batch samples are from the majority group (aligned samples) and $\frac{c}{2}$ from minority (conflicting samples).

- Weighting step: we compute the loss $\mathcal{L}_{sw}$ over the sampled batch as follows:

$$\mathcal{L}_{sw} = w \cdot \mathcal{L}_A + (1-w) \cdot \mathcal{L}_C, w = \frac{2 - c \cdot (1+\beta)}{2} \quad (20)$$

where $\mathcal{L}_A$ is computed over aligned samples in the 'SW' batch and $\mathcal{L}_C$ is computed over the conflicting samples.

Let's compute the variance of the two losses:

$$Var(\mathcal{L}_{sw}) = w^2 \cdot Var(\mathcal{L}_A^{1-c/2}) + (1-w)^2 \cdot Var(\mathcal{L}_C^{c/2}) + 2 \cdot w \cdot (1-w) \cdot Cov(\mathcal{L}_A^{1-c/2}, \mathcal{L}_C^{c/2}) \quad (21)$$

where $Var(\mathcal{L}^m)$ means the variance computed over $m \cdot B$ samples where B is the batch size. For simplicity, we refer to $Var(\mathcal{L}^1)$ as $Var(\mathcal{L})$

Following the fourth assumption, when the model overfits on the conflicting samples (i.e. $\mathcal{L}_C \approx 0$), we can approximate both $Var(\mathcal{L}_C), Cov(\mathcal{L}_A, \mathcal{L}_C)$ to zero. Therefore:

$$Var(\mathcal{L}_{sw}) \approx w^2 \cdot Var(\mathcal{L}_A^{1-c/2}) = \frac{w^2}{1-\frac{c}{2}} \cdot Var(\mathcal{L}_A) = (\frac{2 - c \cdot (1+\beta)}{2})^2 \cdot \frac{2}{2-c} \cdot Var(\mathcal{L}_A) \quad (22)$$

From Eq. (16):

$$\begin{aligned} Var(\mathcal{L}_{ext}) &= (1+\beta)^2 \cdot Var(\mathcal{L}_{lb}) + \beta^2 \cdot Var(\mathcal{L}_b) \\ &\quad - 2 \cdot (1+\beta) \cdot \beta \cdot Cov(\mathcal{L}_{lb}, \mathcal{L}_b) \\ &\geq ((1+\beta) \cdot \sqrt{Var(\mathcal{L}_{lb})} - \beta \cdot \sqrt{Var(\mathcal{L}_b)})^2 \end{aligned} \quad (23)$$

Note that the the inequality reduces to an equality in Eq. (23) if $Cov(\mathcal{L}_{lb}, \mathcal{L}_b) = \sqrt{Var(\mathcal{L}_{lb})} \cdot \sqrt{Var(\mathcal{L}_b)}$.
The covariance term (Cov) can be controlled by the number of shared samples between the biased and less biased batches. If all the aligned samples in the less biased batch are included in the sampled biased batch (i.e. the less biased batch is created by replacing some aligned samples by conflicting ones), we get maximum value for the Cov.

From Eq. (18):

$$Var(\mathcal{L}_{lb}) \approx (1-\frac{c}{2})^2 \cdot Var(\mathcal{L}_A^{1-c/2}) = (1-\frac{c}{2}) \cdot Var(\mathcal{L}_A) \quad (24)$$

And from Eq. (17)

$$Var(\mathcal{L}_b) \approx Var(\mathcal{L}_A) \quad (25)$$

Finally, substituting Eq. (24) and Eq. (25) in Eq. (23):

$$Var(\mathcal{L}_{ext}) \geq ((1+\beta) \cdot \sqrt{1-\frac{c}{2}} - \beta)^2 \cdot Var(\mathcal{L}_A) \quad (26)$$

According to the fourth assumption, we are interested in the range where $c \cdot (\beta + 1) \geq 1$. Using the limits of β defined in Eq. (9), we obtain $\beta \in \left[\frac{1-c}{c}, \frac{2-c}{c}\right]$. As $\beta \to \frac{2-c}{c}$, the representation of aligned samples simulates a vanishing representation (according to Eq. (8)) in the sampled batches, which leads to $\mathcal{L}_A > 0$. Assuming a limited and non-vanishing variance $Var(\mathcal{L}_A)$ (i.e. the model outputs a non-constant prediction for samples from the majority group), we have:

$\beta \to \frac{2-c}{c} \implies Var(\mathcal{L}_{sw}) \approx 0, Var(\mathcal{L}_{ext}) \neq 0$ for $c \in (0, 1]$. This non-vanishing variance of GERNE's loss, if controlled with tuning β to ensure stability, gives the model the chance of escape sharp minima similar to gradient random perturbation [1] and therefore, improve generalization [23, 33, 34].

C. Bounding β

We aim to simplify the upper and lower bounds of β in Eq. (9). We start by simplifying the upper bound:

$$\min_{\substack{(y,a)\in\mathcal{G} \\ \alpha_{ya} \neq \frac{1}{A}}} \max(i^1_{ya}, i^2_{ya})$$

Where: $i^1_{ya} = -\frac{\alpha_{ya}}{c \cdot (\frac{1}{A} - \alpha_{ya})} - 1, i^2_{ya} = \frac{1-\alpha_{ya}}{c \cdot (\frac{1}{A} - \alpha_{ya})} - 1$.
under the constraints that $\forall y \in \mathcal{Y}, \sum_a \alpha_{ya} = 1$, $\alpha_{ya} \in]0, 1[$, $A \geq 2$, and $c \in [0, 1]$. There is at least....

Step 1: Comparing i^1_{ya} and i^2_{ya}

We note that i^1_{ya} is a decreasing function in α_{ya}, and i^2_{ya} is an increasing function in α_{ya}. We can show that if $\alpha_{ya} < \frac{1}{A}$, then $i^2_{ya} > i^1_{ya}$, and if $\alpha_{ya} > \frac{1}{A}$, then $i^1_{ya} > i^2_{ya}$. We conclude by the following:

$$\min_{\substack{(y,a)\in\mathcal{G} \\ \alpha_{ya} < \frac{1}{A}}} \max(i^1_{ya}, i^2_{ya}) = i^2_{y'a'}, \min_{\substack{(y,a)\in\mathcal{G} \\ \alpha_{ya} > \frac{1}{A}}} \max(i^1_{ya}, i^2_{ya}) = i^1_{y''a''}$$

where:

$$\alpha_{y'a'} = \min_{(y,a)\in\mathcal{G}} \alpha_{ya} < \frac{1}{A}, \alpha_{y''a''} = \max_{(y,a)\in\mathcal{G}} \alpha_{ya} > \frac{1}{A}$$

Step 2: Finding the Worst-Case (y, a) Pair

Since $\sum_k \alpha_{y'k} = 1$, we have:

$$\sum_{k \neq a'} \alpha_{y'k} = 1 - \alpha_{y'a'},$$

and hence there exists $j \neq a'$ such that:

$$\alpha_{y'j} \geq \frac{1 - \alpha_{y'a'}}{A-1}.$$

Because $\alpha_{y'a'} < \frac{1}{A}$ and $A \geq 2$, we get:

$$\frac{1 - \alpha_{y'a'}}{A-1} > \frac{1}{A},$$

so:

$$\alpha_{y'j} > \frac{1}{A}.$$

Therefore, $\max(i^1_{y'j}, i^2_{y'j}) = i^1_{y'j} = -\frac{\alpha_{y'j}}{c\cdot(\frac{1}{A}-\alpha_{y'j})} - 1$. Since i^1_{ya} is a decreasing function in $\alpha_{ya} \in]\frac{1}{A}, 1]$, we have $i^1_{y'j} \leq -\frac{\frac{1-\alpha_{y'a'}}{A-1}}{c\cdot(\frac{1}{A}-\frac{1-\alpha_{y'a'}}{A-1})} = \frac{1-\alpha_{y'a'}}{c\cdot(\frac{1}{A}-\alpha_{y'a'})} = i^2_{y'a'}$. Therefore: $i^1_{y'j} \leq i^2_{y'a'}$ and we have $i^1_{y''a''} \leq i^1_{y'j}$. Therefore, the upper bound of β is determined by $i^1_{y''a''}$. For the lower bound, we can simply choose $\beta = -1$ as lower bound ($\beta = -1$ satisfies Eq. (9) and simulates ERM training as in Sec. 4.1.3). Finally, we limit $\beta \in [-1, i^1_{y''a''}]$ later in the experiments in the known attribute case. For the derived upper bound, we can see that $\beta_{\max} = i^1_{y''a''}$ is inversely proportional to c, A.

D. Algorithm 2.

Algorithm 2 GERNE for the unknown attribute case

Input: $\mathcal{X}_y \subseteq \mathcal{X}$ for $y \in \mathcal{Y}$, f with initial $\theta = \theta_0, \tilde{\theta} = \tilde{\theta_0}$ (parameters of the biased model $\tilde{f}$), # epochs E, batch size per class label B, # classes K, # attributes $\tilde{A} = 2$, learning rate η.

1: Training $\tilde{f}$ on biased batches with class balanced accuracy CBA $= \frac{1}{K}\sum_{y\in\mathcal{Y}} \mathbb{P}_{x|y}(y = \arg\max_{y'\in\mathcal{Y}} \tilde{f}_{y'}(x))$ as the evaluation metric for model selection.
2: Select a threshold t and create the pseudo-groups $\tilde{\mathcal{G}}$ by following the steps in Sec. 4.2.
3: Follow **Algorithm 1.** with: $\mathcal{G} \leftarrow \tilde{\mathcal{G}}$.

E. Proposition 1.

Creating both biased and less biased batches using the pseudo-groups $\tilde{\mathcal{G}}$, and with β as hyperparameter, we can simulate batches with a more controllable conditional attribute distribution. Specifically, for $(y, a) \in \mathcal{G}$, we can achieve scenarios where $p_{ext}(a|y) > \max_{\tilde{a}\in\tilde{\mathcal{A}}} p(a|\tilde{a}, y)$ or $p_{ext}(a|y) < \min_{\tilde{a}\in\tilde{\mathcal{A}}} p(a|\tilde{a}, y)$ as opposed to Eq. (13).

Proof. We define $\tilde{\alpha}_{y\tilde{a}}$ the same way as in Eq. (2) for the created pseudo-groups: $\tilde{\alpha}_{y\tilde{a}} = \frac{|\mathcal{X}_{y,\tilde{a}}|}{|\mathcal{X}_y|}$. For a constant c, we create the less biased batch as in Eq. (4):

$$p_{lb}(\tilde{a}|y) = \tilde{\alpha}_{y\tilde{a}} + c\cdot(\frac{1}{2} - \tilde{\alpha}_{y\tilde{a}}). \tag{27}$$

Similar to Eq. (8), the conditional attribute distribution $p_{ext}(\tilde{a}|y)$ is given by:

$$p_{ext}(\tilde{a}|y) = \tilde{\alpha}_{y\tilde{a}} + c\cdot(\beta+1)\cdot(\frac{1}{2} - \tilde{\alpha}_{y\tilde{a}}). \tag{28}$$

We can write $p_{ext}(a|y)$ as follows:

$$p_{ext}(a|y) = \sum_{\tilde{a}\in\tilde{\mathcal{A}}} p_{ext}(\tilde{a}|y)\cdot p(a|\tilde{a}, y). \tag{29}$$

Placing Eq. (28) in Eq. (29), we get

$$p_{ext}(a|y) = \sum_{\tilde{a}\in\tilde{\mathcal{A}}} \tilde{\alpha}_{y\tilde{a}}\cdot p(a|\tilde{a}, y) + c\cdot(\beta+1)\cdot(\frac{1}{2} - \tilde{\alpha}_{y\tilde{a}})\cdot p(a|\tilde{a}, y). \tag{30}$$

For $p(a|\tilde{a} = 1, y) \neq p(a|\tilde{a} = 2, y)$ and $\tilde{\alpha}_{y1} \neq \frac{1}{2}$, to make $p_{ext}(a|y) = p$ for some $p \in [0, 1]$, we can choose:

$$\beta = \frac{p - \sum_{\tilde{a}\in\tilde{\mathcal{A}}} \tilde{\alpha}_{y\tilde{a}}\cdot(a|\tilde{a}, y)}{\sum_{\tilde{a}\in\tilde{\mathcal{A}}} c\cdot(\frac{1}{2} - \tilde{\alpha}_{y\tilde{a}})\cdot(a|\tilde{a}, y)} - 1.\blacksquare \tag{31}$$

Discussion. When $\tilde{\alpha}_{y1} = \frac{1}{2}$, our algorithm is equivalent to sampling uniformly from $\mathcal{X}_y$ and equally from classes. When $p(a|\tilde{a} = 1, y) = p(a|\tilde{a} = 2, y)$, it implies that $\tilde{f}$ has distributed the samples with attribute a and class y equally between the two pseudo-groups. However, in practice, this is precisely the scenario that $\tilde{f}$ is designed to avoid. Specifically, if a represents the presence of spurious attributes (i.e., the majority group), it is likely that $p(a|\tilde{a} = 1, y) < p(a|\tilde{a} = 2, y)$. Conversely, when a represents the absence of spurious features (i.e., the minority group), we would expect $p(a|\tilde{a} = 1, y) > p(a|\tilde{a} = 2, y)$. In fact, $\tilde{f}$ is explicitly trained to exhibit a degree of bias, which inherently disrupts the equality above.

F. Implementation Details

F.1. Implementation details on Datasets-1

For the C-MNIST [3, 27], we deploy a multi-layer perceptron (MLP) with three fully connected layers, while for C-CIFAR-10 [15, 31] and bFFHQ [22, 27], we employ a pretrained ResNet-18 model [14] (pretrained on ImageNet1K [8]) as the backbone. The Stochastic Gradient Descent (SGD) optimizer, with a momentum of 0.9 and a weight decay of 0.01, is applied across all three datasets. Batch sizes are configured as follows: 100 per group/pseudo-group for C-MNIST and C-CIFAR-10, and 32 for bFFHQ. Learning rates are set to 0.1 for C-MNIST in the known attribute case and 0.01 in the unknown attribute case. For C-CIFAR-10 and bFFHQ, the learning rate is set to 0.0001.

For GERNE in the known attribute case, we present results from two experimental configurations. In the first experiment, we set $c = 1$ and $\beta = 0$, which corresponds to resampling [16] from groups, i.e., training on $\mathcal{L}_{lb}$ without extrapolation. In the second experiment, β is tuned for $c \in \{\frac{1}{2}, 1\}$. In the unknown attribute case, t is an additional hyperparameter to be tuned. We avoid using any data augmentations as certain transformations can unintentionally fail to preserve the original label. For example, flips and rotations

in C-MNIST can distort labels (e.g., a rotated "6" appearing as a "9") [41]. For training $\tilde{f}$ in case of unknown attributes in the training set, we employ the same model architecture as f, with modifications to the hyperparameters: the weight decay is doubled, and the learning rate is reduced to one-tenth of the learning rate used to train f. The loss function used is the cross-entropy loss in all the experiments.

F.2. Implementation details on Datasets-2

To ensure a fair comparison of GERNE with other methods in [50], we adhere to the same experimental settings. For the Waterbirds [46] and CelebA [30] datasets, we utilize a pretrained ResNet-50 model [14] as the backbone, while for CivilComments [5], we use a pretrained BERT model [9]. Each backbone is followed by an MLP layer with K output neurons. We employ SGD with a momentum of 0.9 and a weight decay of 0.01 for Waterbirds and CelebA, while for CivilComments, we use AdamW [24] optimizer with a weight decay of 0.0001 and a tunable dropout rate. We set batch sizes to 32 for both Waterbirds and CelebA and 5(16) per group(pseudo-group) for CivilComments. The learning rates are configured as follows: 0.0001 for Waterbirds and CelebA, and 0.00001 for CivilComments. Additionally, we set the bias reduction factors c to 0.5 for Waterbirds and CelebA and to 1 for CivilComments. For image datasets, we resize and center-crop the images to 224×224 pixels. In the case of unknown attributes in the training set, $\tilde{f}$ has the same architecture as f, but we adjust the hyperparameters: the weight decay is doubled, and the learning rate is reduced to one-tenth of the value used to train f. We employ the Cross-entropy loss as the loss function across all experiments. For experiments with unknown attributes in both the training and validation sets, we limit the hyperparameter t search space to the interval $[0, \frac{1}{2}]$.

G. Evaluating GERNE under limited attribute information

To further demonstrate the effectiveness of GERNE in scenarios with limited access to samples with attribute information, we conduct two experiments on the CelebA dataset. In these experiments, we exclude the training set and only use the validation set with its attribute information for training. We follow the same settings and implementation details described earlier. As part of the implementation, we first tune the hyperparameters using the designated evaluation metric. Once we determine the optimal hyperparameters, we fix them and train the model f three times with different random seeds. Finally, we report the average worst-group test accuracy and standard deviation across these runs.

Experiment 1 - Evaluation on test set. In this experiment, f is trained using the full validation set. The worst-group accuracy on the test set is used as the evaluation metric. This setup represents the best possible performance achievable when relying solely on the validation set for training.

Experiment 2 - Cross-validation. we divide the validation set into three non-overlapping folds, ensuring that each fold preserves the same group distribution as the original validation set (i.e., we randomly and equally distribute samples from each group across the folds). We use two folds to train f and reserve the remaining fold for hyperparameter tuning and model selection, using the worst-group accuracy on this fold as the evaluation metric. We repeat this process three times, with each fold serving as the validation fold exactly once. We summarize the average worst-group test accuracy and standard deviation across all nine runs (three folds × three seeds) in Tab. 3.

We also compare these results with DFR, a method that trains the last layer of the model on the validation set after performing ERM training on the training set. GERNE consistently achieves state-of-the-art results, demonstrating its robustness and effectiveness even with severely limited attribute information.

Table 3. Performance Comparison of GERNE and DFR using the validation set for training.

Method	WGA on test set(%)
DFR	86.30 ± 0.30
GERNE - Evaluation on test set	90.97 ± 0.35
GERNE - Cross-validation	88.63 ± 0.59

H. GERNE versus the resampling method

In Tab. 1, GERNE achieves higher GBA compared to the special case of GERNE with $c = 1, \beta = 0$ (resampling method [16]) for both C-MNIST and C-CIFAR-10 datasets. Our explanation behind GERNE superior performance is that resampling method tend to present the majority and minority groups equally in the batch, and the model f tends to prioritize learning the easy-to-learn spurious features associated with the majority group. For instance, the color in C-MNIST. While GERNE undermines learning the spurious correlations by directing the learning process more in the debiasing direction thanks to the extrapolation factor.

I. Additional benchmarks and baselines

We compare GERNE with two new additional baselines: DeNetDM (Sreelatha et al., 2024) and BiasEnsemble (Lee et al., 2023), after adapting GERNE to match their implementation details (e.g., data augmentation). In **??**, we report results on two synthetic datasets (C-MNIST, C-CIFAR-10) and two real-world datasets (Dogs&Cats, newly added; and bFFHQ with an additional bias-conflicting ratio), each under two bias-conflicting ratios. GERNE outperforms BiasEnsemble on both ratios of Dogs&Cats, with a margin of over 10% at the challenging 1% bias-conflicting scenario. GERNE also outperforms or remains competitive

with both baselines on the remaining datasets. These results further demonstrate GERNE's effectiveness as a debiasing method.