Gradient Extrapolation for Debiased Representation Learning

Supplementary Material

A. The Extrapolated Conditional Attribute Distribution of GERNE:

$\mathcal{L}$ 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 $\mathcal{X}_{y,a}$ (i.e., $p(x|y, a) = \frac{1}{|\mathcal{X}_{y,a}|}$). Substituting Eq. (14), Eq. (15) in Eq. (5):

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

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:

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

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 terms 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 that hold the spurious correlation in each class are aligned with the class label (i.e., $|\mathcal{X}_{y,a=y}| > |\mathcal{X}_{y,a\neq y}|$), and the dataset is highly biased (i.e., $\frac{|\mathcal{X}_{y,a\neq y}|}{|\mathcal{X}_{y,a=y}|} \ll 1$).
4. In a highly biased dataset, best performance is coupled with overrepresenting the minority groups (as illustrated in Fig. 2) in early stages of training. Therefore, an overfitting on the minority groups is expected before the overfitting on the majority groups.

We refer to the expected loss of the majority samples as $\mathcal{L}_A$, and the expected loss of the minority 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 samples of the minority group 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 (majority group), 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, analogous to the less biased batch in GERNE, where $(1 - \frac{c}{2})$ of the samples are from the majority group (aligned samples) and $\frac{c}{2}$ from minority group.
- 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 the samples of the majority group in the "SW" batch and $\mathcal{L}_C$ is computed over the samples of the minority group.

Both $\mathcal{L}_{ext}$ and $\mathcal{L}_{sw}$ in Eq. (19), Eq. (20) are equivalent in expectation. Let's compute the variance of the two losses:

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

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 samples of the minority group (i.e., $\mathcal{L}_C \approx 0$), we can neglect both $Var(\mathcal{L}_C), Cov(\mathcal{L}_A, \mathcal{L}_C)$ terms. Therefore:

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

From Eq. (16):

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

Note that 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 biased batch (i.e., the less biased batch is created by replacing some samples of the majority group with samples from the minority group), we then maximize $Cov(.,.)$. From Eq. (18), we can write:

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

and from Eq. (17):

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

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

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

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 the aligned samples is vanishing (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 non-constant predictions for samples from the majority group), we have:

$\beta \to \frac{2-c}{c} \implies Var(\mathcal{L}_{sw}) \approx 0$, while $Var(\mathcal{L}_{ext}) \neq 0$ 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 escaping sharp minima similar to gradient random perturbation [1] and therefore, improves generalization [23, 33, 34].

C. Simplifying the Bounds of 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 $i^1_{ya} = -\frac{\alpha_{ya}}{c\cdot(\frac{1}{A}-\alpha_{ya})} - 1, i^2_{ya} = \frac{1-\alpha_{ya}}{c\cdot(\frac{1}{A}-\alpha_{ya})} - 1$, and under the following constraints: $\forall y \in \mathcal{Y}, \sum_a \alpha_{ya} = 1$, $\alpha_{ya} \in]0,1[\setminus\{\frac{1}{A}\}$, $A \geq 2$, and $c \in (0, 1]$.

We note that i^1_{ya} is a decreasing, and i^2_{ya}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 with 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:

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

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

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

which implies that there exists $j \neq a'$ such that:

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

Given that $\alpha_{y'a'} < \frac{1}{A}$ and $A \geq 2$, it follows that

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

and hence $\alpha_{y'j} > \frac{1}{A}$. Therefore, we have

$$\max(i^1_{y'j}, i^2_{y'j}) = i^1_{y'j} = -\frac{\alpha_{y'j}}{c\cdot(\frac{1}{A}-\alpha_{y'j})} - 1.$$

Since i^1_{ya} is a decreasing function in $\alpha_{ya} \in (\frac{1}{A}, 1]$, we have

$$i^1_{y'j} \leq -\frac{\frac{1-\alpha_{y'a'}}{A-1}}{c \cdot (\frac{1}{A} - \frac{1-\alpha_{y'a'}}{A-1})} - 1 = \frac{1 - \alpha_{y'a'}}{c \cdot (\frac{1}{A} - \alpha_{y'a'})} - 1 = i^2_{y'a'}.$$

Thus: $i^1_{y'j} \leq i^2_{y'a'}$, and since $i^1_{y''a''} \leq i^1_{y'j}$, we conclude that the upper bound of β is: $\beta_{\max} = i^1_{y''a''}$. For the lower bound of β, we can follow the same previous step, but we choose $\beta = -1$ as the lower bound ($\beta = -1$ satisfies Eq. (9) and simulates ERM training as shown in Sec. 4.1.3). In conclusion, for the known attributes case, we tune β within the interval: $[\beta_{\min}, \beta_{\max}] = [-1, i^1_{y''a''}]$. The upper bound $\beta_{\max}$ is inversely proportional to both c, A. Consequently, larger values of c reduce the feasible range for the extrapolation factor, making GERNE appear more sensitive to small variations in β.

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 $\eta$.
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 $\tilde{\mathcal{G}}$: For each class $y$, we compute the predictions $\tilde{y}_i = \text{softmax}(\tilde{f}(x_i))_y$ for each $x_i \in \mathcal{X}_y$. We then form the pseudo-minority group $\mathcal{X}_{y,\tilde{a}=1}$ as the samples $x_i$ that have the smallest $\lfloor t \cdot |\mathcal{X}_y| \rfloor$ predictions. The remaining samples form the pseudo-majority group $\mathcal{X}_{y,\tilde{a}=2} = \mathcal{X}_y \setminus \mathcal{X}_{y,\tilde{a}=1}$.
3: Follow Algorithm 1 with $\tilde{\mathcal{G}}$ replacing $\mathcal{G}$. We consider a higher upper bound for $\beta$ than $\beta_{\max}$ derived in Appendix C as justified by the proof of Proposition 1. in Appendix E.
```

E. Proposition 1.

Creating both biased and less biased batches using the pseudo-groups $\tilde{\mathcal{G}}$, and with β as a 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 $\alpha_{y\tilde{a}}$ the same way as in Eq. (2) for the created pseudo-groups: $\alpha_{y\tilde{a}} = \frac{|\mathcal{X}_{y,\tilde{a}}|}{|\mathcal{X}_y|}$. For a constant c and $A = 2$, we create the less biased batch as in Eq. (4):

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

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

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

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

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

For $p(a|\tilde{a} = 1, y) \neq p(a|\tilde{a} = 2, y)$ and $\alpha_{y1} \neq \frac{1}{2}$, to make $p_{ext}(a|y) = p$ for some $p \in [0, 1]$, we tune β until reaching the β_{target} defined as follows:

$$\beta_{target} = \frac{p - \sum_{\tilde{a}\in\tilde{\mathcal{A}}} \alpha_{y\tilde{a}} \cdot p(a|\tilde{a}, y)}{\sum_{\tilde{a}\in\tilde{\mathcal{A}}} c \cdot (\frac{1}{2} - \alpha_{y\tilde{a}}) \cdot p(a|\tilde{a}, y)} - 1. \tag{31}$$

Discussion. When $\alpha_{y1} = \frac{1}{2}$, 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 $\mathcal{X}_{y,\tilde{a}\in\mathcal{A}}$. However, in practice, this is exactly 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 above equality.

F. Implementation Details

F.1. Implementation Details for 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 ResNet-18 model [14], pretrained on ImageNet1K [8], as the backbone. We apply the Stochastic Gradient Descent (SGD) optimizer with a momentum of 0.9 and a weight decay of 10^{-2} across all three datasets. We set the batch size to 100 per group or pseudo-group for both C-MNIST and C-CIFAR-10, and to 32 for bFFHQ. For C-MNIST, we use a learning rate of 10^{-1} in the known attributes case and

10^{-2} in the unknown attributes case. For C-CIFAR-10 and bFFHQ, we use a learning rate of 10^{-4}.

In the unknown attribute case, we treat the threshold t as an additional hyperparameter that requires tuning. 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 for all the experiments.

F.2. Implementation Details for Datasets-2

To ensure a fair comparison between GERNE and the methods in [50], we adopt the same experimental settings. For Waterbirds [46] and CelebA [30] datasets, we use a pretrained ResNet-50 model [14] as the backbone, while for CivilComments [5], we use a pretrained BERT model [9]. We append an MLP classification head with K outputs. We employ SGD with a momentum of 0.9 and a weight decay of 10^{-2} for Waterbirds and CelebA. For CivilComments, we use AdamW [24] optimizer with a weight decay of 10^{-4} 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: 10^{-4} for Waterbirds and CelebA, and 10^{-5} 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 search space for t to the interval $[0, \frac{1}{2}]$.

G. Evaluating GERNE with Limited Attribute Information

To further demonstrate the effectiveness of GERNE in scenarios with limited access to samples with known attributes, we conduct two experiments on the CelebA dataset using only the validation set with its attribute information for training (excluding the training set). We follow the same experimental setup and implementation details outlined in Appendix F.2. 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.

Evaluation on the Test Set. In this experiment, we train the model f using the entire validation set and use the worst-group test accuracy as the evaluation metric. This setup represents the best possible performance achievable when relying solely on the validation set for training.

Cross-Validation. In this experiment, we divide the validation set into three non-overlapping folds, ensuring that each fold preserves the same group distribution as the original validation set. Specifically, we randomly and evenly distribute samples from each group in the validation set across the folds. We train f using two of the folds, and use the remaining fold for hyperparameter tuning and model selection, where the worst-group accuracy on this fold serves as the evaluation metric. We repeat this process three times so that each fold is used once as the validation fold. Finally, we report the average worst-group test accuracy and standard deviation across all nine runs (three folds × three seeds) in Tab. 3.

We compare the results of GERNE with DFR, a method that trains the final layer on the validation set following ERM training on the training set. GERNE consistently achieves state-of-the-art results, demonstrating its robustness and effectiveness even under limited attribute information.

Table 3. Performance comparison of GERNE and DFR. GERNE uses only the validation set for training. We report the worst-group test accuracy (%) and standard deviation over three trials.

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