IMPROVED GROUP ROBUSTNESS VIA CLASSIFIER RETRAINING ON INDEPENDENT SPLITS

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep neural networks learned by minimizing the average risk can achieve strong average performance, but their performance for a subgroup may degrade, if the subgroup is underrepresented in the overall data population. Group distributionally robust optimization (Sagawa et al., 2020a, GDRO) is a standard baseline for learning models with strong worst-group performance. However, GDRO requires group labels for every example during training and can be prone to overfitting, often requiring careful model capacity control via regularization or early stopping. When only a limited amount of group labels is available, Just Train Twice (Liu et al., 2021, JTT) is a popular approach which infers a pseudo-group-label for every unlabeled example. The process of inferring pseudo labels can be highly sensitive during model selection. To alleviate overfitting for GDRO and the pseudo labeling process for JTT, we propose a new method via classifier retraining on independent splits (of the training data). We find that using a novel sample splitting procedure achieves robust worst-group performance in the fine-tuning step. When evaluated on benchmark image and text classification tasks, our approach consistently reduces the requirement of group labels and hyperparameter search during training. Experimental results confirm that our approach performs favorably compared with existing methods (including GDRO and JTT) when either group labels are available during training or are only available during validation.

1 INTRODUCTION

For many tasks, deep neural networks (DNNs) are often developed where the test distribution is identical to and independent of the train distribution, which can be referred to as IID generalization. The performance of a DNN is also known to worsen when the testing distribution differs from the training distribution. This problem is often referred to as out-of-distribution (OOD) generalization. OOD generalization is crucial in safety-critical applications such as self-driving cars (Filos et al., 2020) or medical imaging (Oakden-Rayner et al., 2020). Hence, addressing the problem of OOD generalization is foundational for real-world deployment of deep learning models.

A notable setting where OOD generalization problems appear is the *group-shift* setting, where different groups of the data may have a distribution shift. In this setting, there are predefined attributes that divide the input space into different groups of interest. Here, the goal is to find a model that performs well across several predefined groups (Sagawa et al., 2020a). Similar to other problems in OOD generalization, DNNs learned by empirical risk minimization (ERM) are observed to suffer from poor worst-group performance despite good average-group performance.

The difficulty with learning group robust DNNs can be attributed to the phenomenon of shortcut learning (Geirhos et al., 2020) or spurious correlation (Sagawa et al., 2020a; Arjovsky et al., 2019). Shortcut learning poses that ERM favors those models that discriminate based on simpler and/or spurious features of the data. However, one wishes for the learning algorithm to produce a model that uses features (i.e. correlations) that performs well not only on the train distribution, but also on all potential distributions that a task may generate, like that of a worst-group distribution.

In recent years, the group-shift setting has received considerable attention, where Sagawa et al. (2020a) first investigates distributional robust optimization (DRO) (Duchi et al., 2021; Ben-Tal et al., 2013) in this setting and introduces *Group DRO* (GDRO) that attempts to directly optimize for the worst-group error. Since then, GDRO has been the standard method for producing group-robust

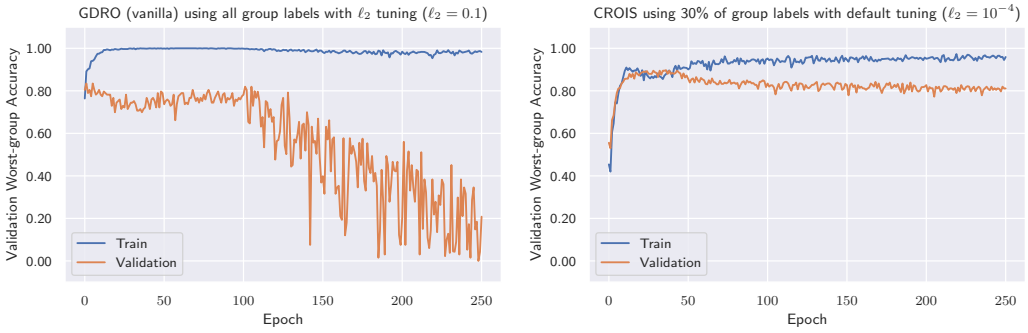

Figure 1: Worst-group learning curve on the Waterbird dataset between GDRO (**left**) and CROIS (**right**) from the setting in Section 4.2. In the left figure, the validation accuracy of GDRO becomes unstable as the number of epochs increase beyond 100 while the training accuracy remains close to 100%. Our approach CROIS instead uses 30% of the training data with group labels for fine-tuning the classification layer of a DNN that is obtained via ERM using the rest of the training data without group labels (see also Algorithm 1). This allows CROIS to reuse ERM features while improving generalization compared with GDRO.

models. However, GDRO can be sensitive towards model capacity (Sagawa et al., 2020a) and require group labels for all examples at training time.

As group annotations can be expensive to obtain, many works consider ways to reduce the amount of groups labels needed (Liu et al., 2021; Creager et al., 2021; Nam et al., 2022). These methods usually follow the framework of first inferring pseudo-group-labels using a certain referenced model (pseudo-labeling) and then applying a group-robust algorithm like GDRO on the pseudo-labelled data. While results for these methods have been promising, they usually introduce several more sensitive hyperparameters that can be expensive to tune. Our work aims to obtain group robust models using as few group labels as possible while alleviating the need to carefully control model capacity.

Our contribution. In this work, we propose a simple approach, called CROIS. By foregoing the error-prone and costly pseudo-labeling phase to instead concentrate on efficiently utilizing group labels by applying them *only* to the final classifier layer, CROIS achieves good robust performance *without* relying on a multitude of hyperparameters and large scale tuning, which has been a growing concern in the community (Gulrajani and Lopez-Paz, 2021).

In short, CROIS takes advantage of the features learned by ERM (Kang et al., 2019; Menon et al., 2021) while overcoming the deficiency of its memorization behavior (Sagawa et al., 2020b) by utilizing the training data as two independent splits: one *group-unlabeled* split to train the feature extractor and one *group-labeled* split to retrain only the classifier with a robust algorithm like GDRO. We demonstrate through ablation studies that the use of independent splits is crucial for robust classifier retraining. Furthermore, CROIS's restriction of GDRO to only a low-capacity linear layer reduces GDRO's sensitivity towards model capacity as well as the amount of data needed for GDRO to generalize well (e.g. Figure 1 and Figure 3).

For various settings where group labels are only partially available during training (Section 4.1), our experimental results on standard datasets including Waterbird, CelebA, MultiNLI, and CivilComments show improved performance over existing methods including JTT (Liu et al., 2021) and SSA (Nam et al., 2022), despite minimal parameter tuning and no reliance on pseudo labeling. In another setting where more group labels are available (Section 4.2), even when using only a fraction of the available group labels, our competitive robust performance against GDRO demonstrates our method's label efficiency. Finally, our results provide further evidences of ERM trained DNNs containing good features on both image classification (Menon et al., 2021) and natural language classification tasks.

1.1 Related works

There are three main settings for the group-shift problem: (1) full availability of group labels, (2) limited availability of group labels, and (3) no availability of group label. Other related areas include domain generalization and long-tailed classification.

Full training group labels. Most methods here revolve around up-weighing minority groups, subsampling minority groups (Sagawa et al., 2020b), or performing GDRO (Sagawa et al., 2020a). Follow-up works include integrating data augmentation via generative model (Goel et al., 2020) or selective augmentation (Yao et al., 2022) to a robust training pipeline.

Limited access to group labels. In this setting, the approach of inferring more group labels for the group-unlabeled data remains the most popular. These *pseudo-group-labels* are usually generated by training a referenced model that performs the labeling. For example, (Liu et al., 2021) utilizes a low-capacity model that create groups by labeling whether an example is correctly classified by the referenced model or not. Similarly, works like (Creager et al., 2021; Dagaev et al., 2021; Krueger et al., 2021; Nam et al., 2022) and (Nam et al., 2020) are variants of this approach of inferring pseudo-group-labels. These methods then proceed to use a group-robust algorithm like GDRO (Sagawa et al., 2020a) or Invariant Risk Minimization (IRM) (Arjovsky et al., 2019) to perform robust training on a new network with the newly generated pseudo-group-labels.

No group labels. This setting removes the ability to validate as well as knowledge of potential groups. This makes the problem more difficult as it is unclear which correlation to look for during training. Some theoretical works in this space include Hashimoto et al. (2018); Lahoti et al. (2020). Sohoni et al. (2020) proposes a popular empirical approach in this setting and has popularized the pseudo-labeling and retraining approach. This setting is related to *domain generalization*. Gulrajani and Lopez-Paz (2021) shows through mass-scale experiments that most OOD generalization methods do not improve over ERM given the same amount of tuning and model selection criterion.

Long-tailed classification. The long-tailed problem concerns with certain classes having significantly fewer training examples than others (see for example Haixiang et al. (2017) or Zhang et al. (2021) for a survey). Some techniques from the long-tail literature, like Cao et al. (2019), has been applied to the group-shift setting to account for the groups imbalances, as in Sagawa et al. (2020a). Insights from applications of representation learning in the long-tail problem (Kang et al., 2019) gives valuable evidences for ERM trained DNNs containing good features which is central for our method.

2 PRELIMINARIES

For a classification task $\mathcal{T}$ of predicting labels in $\mathcal{Y}$ from inputs in $\mathcal{X}$, we are given training examples $\{(x_i, y_i)\}_{i=1}^n$ that are drawn IID from some train distribution $\mathcal{D}_{\text{train}}$. In the domain generalization setting, we want good performance on some unknown test distribution $\mathcal{D}_{\text{test}}$ that is different but related to $\mathcal{D}_{\text{train}}$ through the task $\mathcal{T}$. More explicitly, we wish to find a classifier f from some hypothesis space $\mathcal{F}$ using $\mathcal{D}_{\text{train}}$ such that the classification error $L(f) = \mathbb{E}_{x,y\sim\mathcal{D}_{\text{test}}}[f(x) \neq y]$ of f w.r.t. $\mathcal{D}_{\text{test}}$ is low. This framework encapsulates many problems like adversarial robustness, domain adaptation, long-tail learning, few-shot learning, and the problem considered here: group-shift.

In the group-shift setting (Sagawa et al., 2020a), we further assume that associated with each data point x is an *attribute* $a(x)$ (some sub-property or statistics of x) from a set of possible attributes $\mathcal{A}$. These attributes along with the labels form the set of possible groups $\mathcal{G} = \mathcal{A} \times \mathcal{Y}$ that each example can take. We denote an input x's group label as $g(x) \in \mathcal{G}$. We then define the classification error of a predictor f (w.r.t. a fixed implicit distribution) restricted to a group $g \in \mathcal{G}$ to be $L_g(f) := \mathbb{E}_{x,y|g(x)=g}[f(x) \neq y]$. The notion of *worst-group error* upper bounds the error of f w.r.t. any group $L_{wg}(f) := \max_{g\in\mathcal{G}} L_g(f)$. Using this notation, the group-shift problem aims to discover a classifier in $\arg\min_{f\in\mathcal{F}}\{L_{wg}(f)\} = \arg\min_{f\in\mathcal{F}}\{\max_{g\in\mathcal{G}} L_g(f)\}$. Here, we see that the group-shift problem is just a special case of the domain generalization problem when $\mathcal{D}_{\text{test}}$ is the distribution consisting of only the points (x, y) with $g(x)$ being restricted to the worst-group of f in $\mathcal{G}$. Here, GDRO solves this objective by performing a minimax optimization procedure that alternates between the model's weight and relaxed weights on the groups.

Spurious Correlations and Memorization. As an example, consider the Waterbird dataset as in Sagawa et al. (2020a), where the task is to distinguish whether an image of a bird is a waterbird or a landbird. The bird images are then modified with either a water or land background. In terms of our problem, the type of bird forms the labels $\mathcal{Y}$, and the backgrounds are set to be the attribute $\mathcal{A}$ for each type of bird. Altogether, these form four groups: $\mathcal{G} = \mathcal{Y}\times\mathcal{A} = \{\text{waterbird, landbird}\}\times\{\text{water, land}\}$.

This dataset is constructed so that the proportion of birds on matching backgrounds (i.e. waterbird on water background and landbird on land background) is significantly more than those of the

Algorithm 1 Classifier Retraining on Independent Splits (CROIS)

1: **Input:** Training data D_L with group labels, training data without group labels D_U, and validation data D_{val} with group labels. Classifier retraining algorithm $\mathcal{R}$ (default to GDRO).
2: Split proportion p: Partition D_L into two parts, D_1 and D_2, each with proportion $|D_1| = (1-p) \cdot |D_L|$ and $|D_2| = p \cdot |D_L|$. Set $D_U \leftarrow D_U \cup D_1$. Set $D_L \leftarrow D_2$.
3: Obtain the initial model f by running ERM on D_U and selecting the best model in terms of average accuracy on D_{val}.
4: Perform classifier retraining $\mathcal{R}$ with feature extractor f on D_L and then select the best model in terms of worst-group accuracy on D_{val} as the final output.

mismatched backgrounds. This is so that the backgrounds could be spuriously correlated with the labels, as predicting the background alone would achieve a high on average accuracy w.r.t. the train distribution already. As expected, for ERM trained models, the groups with the highest error are the minority groups where the background mismatches the type of the bird, suggesting that the model is actually predicting using the background instead of the bird. Furthermore, the fact that these high-capacity models achieve *zero* training error leads to the conclusion that these models not only utilize spurious features like the background to make its predictions, but also must have memorized the minority groups during its training process (Sagawa et al., 2020b). These problems are common when there is data imbalance Feldman and Zhang (2020). In the next section, we propose a method that attempts to circumvent these issues.

3 METHOD: CLASSIFIER RETRAINING ON INDEPENDENT SPLITS

Algorithm 1 presents an outline for our main method: *Classifier Retraining On Independent Splits* or *CROIS*. CROIS is simple: (1) partition/organize the training data into one *group-labeled* split and one *group-unlabeled* split, (2) obtain an ERM trained feature extractor with the group-unlabeled split, and (3) perform robust classifier retraining with the group-labeled split, where classifier retraining refers to fine-tuning the final linear layer of a DNN. The optional parameter p in step 2 determines the amount of group labels used at train time so that D_U is not too small. This option will be utilized when we compare CROIS against GDRO in Section 4.2 .

Motivation. Inspired by previous works that have demonstrated the potential of simple ERM trained DNNs on a variety of OOD tasks (Gulrajani and Lopez-Paz, 2021; Rosenfeld et al., 2021) and long-tailed tasks (Kang et al., 2019), our method focuses on developing a simple method around ERM trained DNNs. As soon to be discussed, a *strength* of an ERM trained DNN is that there are convincing evidences that it contains *good features*. On the other hand, its *weaknesses* involve memorizing examples and using spurious features. CROIS is designed to alleviate these weaknesses while taking advantage of the good features of an ERM trained DNN.

Models trained by ERM contains good features. While not the first work to notice that ERM trained models contain good features, Kang et al. (2019) demonstrates this hypothesis through extensive experiments on several long-tailed vision datasets to investigate different learning strategies for obtaining a feature extractor as well as ways to fine-tune the classifier layer. There, an ERM trained feature extractor combined with a non-parametric method of rescaling[1] the classifier layer achieves (then) state-of-the-art results on all three datasets. A similar study (Menon et al., 2021) has been done to confirm this hypothesis on vision datasets in the group-shift setting. This suggests that one key to the data-imbalance problem (for which the group-shift problem almost always suffers) lies in correcting the classifier layer, which forms the first phase of CROIS. It is peculiar that rescaling the classifier works best whereas intuition suggests a data-dependent method like classifier retraining would be better. We hypothesize that this is related to the next issue of our discussion.

Memorization behavior of high capacity DNNs. It has now been well known of high capacity DNNs' ability to memorize training examples (Zhang et al., 2017). In the group-shift setting, this behavior has been investigated by Sagawa et al. (2020b), which provides empirical and theoretical justifications for DNNs' memorization behavior of minority groups' training examples. This behavior of memorizing minority examples have also been observed in the broader framework of data imbalances (Feldman

[1]Rescaling each row of the linear classifier using the row's norm to some power. See Kang et al. (2019).

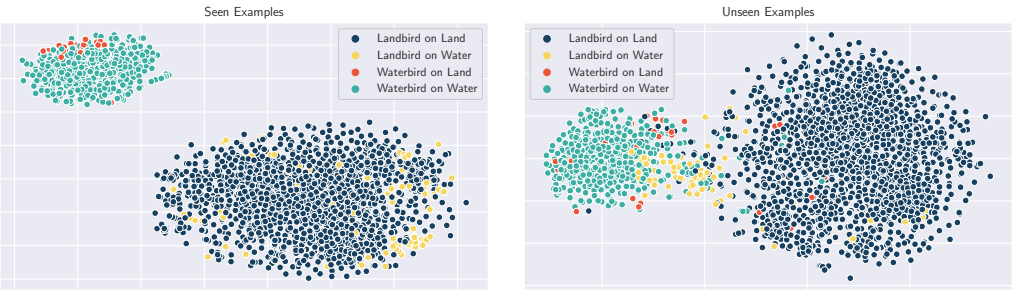

Figure 2: tSNE projection (Van der Maaten and Hinton, 2008) of the features of an ERM trained ResNet50 on *seen* (**left**) versus *unseen* (**right**) examples from Waterbird.[2] The features of the minority groups (orange and yellow) from the *unseen examples* appear better separated from the majority groups than that of the seen examples. Using unseen examples plays a major role in CROIS's ability to improve worst-group performance via robust classifier retraining.

and Zhang, 2020). One way to circumvent memorization is to control the model's capacity by incorporating some combinations of high ℓ_2 regularization, early stopping, and other correctional parameters as has been done in Sagawa et al. (2020a). However, the additional tuning required might be costly. We instead tackle this via *independent splits*.

Circumventing memorization with independent splits. Examples being memorized must inevitably impact their ability to be useful in subsequent usage. As memorized examples' (i.e. already correctly classified) loss must be low, their gradients contain little information to be of much use. Furthermore, the features of memorized examples might not be representative of their group during test time: Figure 2 presents a visualization of the features between seen versus unseen examples. Thus, combining this observation with the evidences for ERM trained DNNs containing good features, CROIS performs robust classifier retraining on *unseen* examples (D_L in Algorithm 1) in hope of learning a classifier that utilizes features more representative of examples during test time.

4 Experiments

In this section, we detail the main results of our experiments in two popular group-shift settings: (1) where only group labels from the validation set are available (as in Liu et al. (2021); Nam et al. (2022)); (2) when a fraction of the training dataset examples also have group labels.

Setup. We use a similar setup as Liu et al. (2021). To demonstrate the ease of tuning of CROIS, unless noted otherwise (e.g. Table 2), we *fix* the hyperparameters of both the ERM and the robust classifier retraining phase, using standard parameters for ERM (see Appendix A for full details). Further results of CROIS with tuned hyperparameters are presented in Section C of the Appendix.

Datasets. We experiment on four standard datasets in the group-shift setting:

- **Waterbird** (Sagawa et al., 2020a). Combining the bird images from the CUB dataset (Welinder et al., 2010) with water or land backgrounds from the PLACES dataset (Zhou et al., 2017), the task is to classify whether an image contains a *landbird* or a *waterbird* without confounding with the background. There are 4795 total training examples, where the minority group (*waterbird, land background*) has only 56 examples. In our experiments, we report the weighted test *average accuracy* due to the skewed nature of the val and test set – this is standard per Sagawa et al. (2020a).
- **CelebA** (Liu et al., 2015) is a popular image dataset of celebrity faces. The task is to classify the celebrity in the image is *blond* or *not blond*, with *male* or *not male* as the confounding attribute. There are 162770 total training examples, and the smallest group (*blond, male*) has 1387 examples.
- **MultiNLI** (Williams et al., 2017) is a natural language inference dataset for determining whether a sentence's hypothesis is *entailed* by, is *neutral* with, or *contradicts* its premise. The spurious attribute is the presence of negation words like *no, never*, or *nothing*. This task has 6 groups, with 206175 total training and 1521 in the minority group examples (*is entailed* and *contains negation*).

[2]Half of the data is used to obtain a feature extractor while the other half is used to obtain the features of the unseen examples.

Table 1: Experimental results for the setting when only group labels from the validation set are used. Results for JTT and SSA are taken from Liu et al. (2021) and Nam et al. (2022), respectively.

	Waterbird		CelebA		MultiNLI		CivilComments	
Method	Avg Acc	Wg Acc	Avg Acc	Wg Acc	Avg Acc	Wg Acc	Avg Acc	Wg Acc
JTT	93.9	86.7	88.0	88.1	78.6	72.6	91.1	69.3
SSA	92.2 (0.87)	89.0 (0.55)	92.8 (0.11)	**89.8** (1.28)	79.9 (0.87)	76.6 (0.66)	88.2 (1.95)	69.9 (2.02)
CROIS	92.1 (0.29)	**90.9** (0.12)	91.6 (0.61)	88.5 (0.87)	81.4 (0.06)	**77.4** (1.21)	90.6 (0.20)	**70.3** (0.34)

- **CivilComments-WILDS** (Borkan et al., 2019; Koh et al., 2021) is a natural language dataset where the task is to classify whether a sentence is *toxic* or *non-toxic*. There are 8 demographics – *male, female, white, black, LGBTQ, Muslim, Christian,* and *other religion*– forming 16 groups that *overlap*, because a comment can contain multiple demographics. Following Koh et al. (2021), we evaluate on all 16 groups but only use the attribute *black* along with the label in training. There is a total of 269038 training examples with 1045 minority examples from (*other religion*, *toxic*).

Models. We use ResNet50 (He et al., 2016) with ImageNet initialization and batch-normalization for CelebA and Waterbird. We use pretrained BERT (Devlin et al., 2018) for MultiNLI and CivilComments. We use the original train-val-test split in all the datasets and report the *test* results. Cross-entropy is used as the base loss for all objectives. SGD with momentum 0.9 is used for the vision datasets while the AdamW optimizer with dropout and a fixed linearly-decaying learning rate is used for BERT. We use a batch size of 16 for CivilComments and 32 for the rest of the datasets. We do *not* use any additional data augmentation or learning rate scheduler in our results.

4.1 Result of using group labels from only the validation set

In this section, we consider the setting where only group labels from the validation set are available, and the training set is treated as a group-unlabeled set (Liu et al., 2021; Nam et al., 2022). Most methods in this setting employ some pseudo-labeling approach to generate pseudo-group-labels that are then used to train a *new* network via a robust algorithm like GDRO. On the other hand, CROIS simply uses half of the validation set for classifier retraining and the other half for model selection and does not rely on pseudo-labeling. CROIS also reuses the initial model for the retraining phase, making CROIS closer to that of a 1 phase procedure with additional fine-tuning.

We compare CROIS against JTT (Liu et al., 2021) and SSA (Nam et al., 2022) in Table 1. We report the Test Average (*Avg Acc*) and Worst-Group Accuracy (*Wg Acc*). The results in Table 1 show that CROIS outperforms JTT on all 4 datasets and SSA on 3 datasets using default parameters.

Comparison with JTT. While both JTT and CROIS employ a 2 phase approach, the two methods differ significantly from one another. First, the initial ERM phase of JTT is used to *infer* pseudo-group-labels for the group-unlabeled data, which is then used for full GDRO retraining in the second phase. JTT's first phase requires careful hyperparameter tuning and capacity control using the group-labeled validation set to accurately produce pseudo-group-labels (as noted in section 5.4 of Liu et al. (2021)). On the other hand, CROIS trains a *single* ERM model and simply retrains the last layer with any available group labels. Second, JTT's final performance is limited by GDRO's performance on the full network, which can be worsen by mislabeled pseudo-labels from the first phase. In contrast, we demonstrate in Section 4.2 that, by limiting GDRO to only the last layer, CROIS is competitive to full GDRO even when using only a fraction group labels and minimal tuning.

Comparison with SSA. Similarly to JTT, SSA is also a two-phase approach that performs pseudo-labeling and then robust training (e.g. via GDRO) with the inferred pseudo-labels. In SSA pseudo-labeling phase, it trains a separate network that *predicts the group* rather than the class. By treating the pseudo-labeling problem as a semi-supervised problem along with incorporating threshold correction to calibrate for the pseudo-labels, SSA's pseudo-labeling capability is shown to improve upon JTT. On the other hand, CROIS, despite not relying on pseudo labeling at all, manages to outperform SSA on 3 out of 4 datasets using default parameter.

Adjusting validation dataset size. Following the setup in JTT and SSA, we vary the size of the validation set (20%, 10% and 5% of the original) to test whether our results still hold in these

Table 2: Worst-group test accuracy for reduced validation. Results for SSA and JTT are taken from Table 3 of Nam et al. (2022). Results for CROIS here are *tuned* due to the smaller dataset size.

% of group-labels from the validation set	CelebA				Waterbird			
	100%	20%	10%	5%	100%	20%	10%	5%
JTT (Liu et al., 2021)	81.1	81.1	81.1	82.2	86.7	84.0	86.9	76.0
SSA (Nam et al., 2022)	89.8	88.9	**90.0**	86.7	89.0	88.9	88.9	87.1
CROIS (Ours)	**90.6**	**90.1**	89.4	**88.9**	**90.8**	**90.0**	**90.0**	**88.8**

Table 3: Results for running CROIS on some fraction of the training data. Numbers in parentheses denote one standard deviation across three runs. *NCRT* refers to naive classifier retraining i.e. when independent split is *not used* during the retraining phase. Results marked with † are taken from (Sagawa et al., 2020a). *For Waterbird, we omit the result for $p = 0.05$ due to the small dataset size and not being able to sample any minority-group example for robust retraining.

	Waterbird		CelebA		MultiNLI		CivilComments	
	Avg Acc	Wg Acc	Avg Acc	Wg Acc	Avg Acc	Wg Acc	Avg Acc	Wg Acc
ERM	96.9	69.8	95.6	44.4	82.8	66.0	92.1	63.2
GDRO	93.2†	86.0†	91.8†	88.3†	81.4†	77.7†	89.6 (0.23)	70.5 (2.10)
CROIS' p – group-labeled fraction used for retraining (with $1-p$ unlabeled fraction for the ERM phase)								
0.05	*	*	91.9 (0.50)	88.9 (1.10)	81.8 (0.15)	73.8 (1.54)	90.8 (0.40)	63.3 (7.82)
0.10	95.4 (1.10)	83.5 (3.24)	91.3 (0.36)	90.3 (0.82)	80.8 (0.51)	75.3 (2.06)	89.5 (1.81)	68.7 (1.72)
0.30	90.8 (0.35)	**89.6** (1.15)	91.3 (0.44)	**90.6** (0.95)	80.0 (0.31)	**77.9 (0.17)**	89.7 (0.33)	68.6 (1.53)
0.50	90.4 (0.95)	89.5 (0.59)	91.9 (0.35)	88.2 (2.10)	79.8 (0.26)	74.4 (1.00)	89.5 (0.70)	**71.0** (1.50)
NCRT	96.5	75.2	93.9	69.2	82.3	67.9	90.3	67.6

settings. We consider both the Waterbird and CelebA datasets. Note that for this setting, CROIS must be additionally *tuned* to account for the increased difficulty of the reduced group-labels quantity. Nevertheless, the smaller examples quantity along with just training the last layer makes the extra tuning not too expensive (details and setup are deferred until Section E). From Table 2, we find that CROIS outperforms JTT and SSA on various percentage levels.

4.2 Result of varying fraction of training group labels

In this section, we compare CROIS using some fraction of group labels against GDRO (Sagawa et al., 2020a) using *all* the group labels. In contrast to Section 4.1, the validation set is used *only* for model selection and not for classifier retraining here. Once again, we fix parameters of CROIS to its standard ERM parameter to demonstrate its ease-of-tuning (see Appendix A).

Varying fraction setup. We examine CROIS on different amount of group labels available, via the parameter p. This means that $(1-p)$ fraction of the training set is used to obtain a feature extractor in the first phase (that uses no group label), and the rest p fraction of group labels are used for robust classifier retraining. In practice, p is not a parameter to choose (there's no reason to throw away group labels) but rather is limited by the resources available to obtain group labels. Our experiment simulates this scenario where we obtain group labels for some random fraction of the data. Additionally, to demonstrate the importance of retraining with *unseen* examples, we experiment with robust classifier retraining using the same data from the first phase i.e. *without* independent splits – denoted as *NCRT*. We present the main results in Table 3.

Discussions. In Table 3, CROIS outperforms GDRO on both the vision datasets and yields competitive performance to GDRO on the two language datasets despite using only a fraction of group labels and using default hyperparameters. Our result implies that one can obtain comparable or even better robust performance than GDRO by simply collecting group labels for only 30% of the available training data (modulo validation). One exception is in severe group imbalance case as in CivilComments (the minority group consists of only 0.4% of the dataset). There, a higher fraction of group-labeled data is beneficial to obtain more minority-group examples. Hence, a more efficient sampling method to

include more minority examples (e.g. filter by labels first) would be beneficial in practice. Finally, the results for NCRT also show the importance of using independent split for classifier retraining. [3]

Worst-group robustness and proportion of group-labeled data. From the results across the datasets, allocating more examples towards training the feature extractor (lower p) generally yields higher on-average accuracies. The worst-group error after classifier retraining has a more complex interaction with p, as it depends on both the quality of the feature extractor and the amount of group-labeled examples available to perform classifier retraining. While varying the proportion p in our experiments gives a rough estimate of this tradeoff, we hypothesize that the availability of minority group examples is the most important for obtaining a robust classifier. We further support this intuition with an ablation study in Section D.2 where removing non-minority examples has an insignificant impact on the group-robust performance.

Alleviating GDRO's sensitivity towards model capacity. GDRO's requirement for model capacity control via either ℓ_2 regularization or early stopping is well noted in the literature (Sagawa et al., 2020a). In Table 4, we compare GDRO's and CROIS' sensitivity towards different ℓ_2 regularization. While GDRO's performance on Waterbird is relatively uniform, GDRO is more sensitivity to different ℓ_2 settings on CelebA. When $\ell_2 = 1$ for CelebA, we see that GDRO fails altogether (see Figure 3). On the other hand, CROIS achieves consistent performance across different ℓ_2 setting. CROIS controls the model capacity by limiting GDRO to only the last layer. This alleviates GDRO's tendency to overfit and simplifies parameter tuning (as in Figure 1 for Waterbird).

Table 4: Worst-group test accuracy for GDRO and CROIS ($p = 0.30$) with different ℓ_2 regularization.

	CelebA						Waterbird					
ℓ_2 reg.	0	10^{-4}	10^{-3}	10^{-3}	10^{-1}	1	0	10^{-4}	10^{-3}	10^{-3}	10^{-1}	1
GDRO	81.7	81.7	81.7	83.9	**87.8**	0.00	86.8	86.8	86.8	86.8	**87.1**	86.5
CROIS	90.6	**91.5**	90.0	90.0	90.3	90.0	90.3	**90.6**	90.6	90.6	90.0	88.2

Figure 3: While GDRO often requires high ℓ_2 regularization to avoid overfitting, setting ℓ_2 too high might cause instability in GDRO's minimax optimization procedure (**left**).

4.3 Further ablation studies and discussions

Obtaining a good feature extractor. An ablation study on the effects of different validation accuracies and initial algorithm on the feature extractor's quality (measured by robust performance after classifier retraining) is presented in Section B. Similarly to previous works Kang et al. (2019), we find ERM providing the best features over reweighting or GDRO (of which both require group labels). We find that there's a positive correlation between validation average accuracy and features' quality. This then serves as a proxy for CROIS' model selection criterion in the first phase, which significantly simplifies parameter tuning over other two-phase methods in the group-shift setting.

[3] In Sagawa et al. (2020a), Group Adjustment (GA) (Cao et al., 2019) is observed to improve Waterbird's WG accuracy to 90.5%. We also notice an improvement when GA is incorporated to CROIS and obtain a $90.3\% \pm 0.62$ test WG accuracy. We also observe that, similarly to Sagawa et al. (2020a), GA only works for the synthetic dataset Waterbird but not for CelebA nor MultiNLI.

Table 5: Comparison between reweighting, subsampling, and GDRO as classifier retraining algorithms on top of the same feature extractor ($p = 0.30$).

Retraining Method	Waterbird		CelebA		MultiNLI		CivilComments	
	Avg Acc	Wg Acc	Avg Acc	Wg Acc	Avg Acc	Wg Acc	Avg Acc	Wg Acc
Reweighting	95.2	87.1	92.1	85.0	78.9	67.0	88.3	56.4
Subsampling	95.8	81.1	91.6	86.1	78.6	64.0	91.8	59.3
GDRO	91.4	**90.2**	91.6	**90.4**	80.3	**78.0**	89.7	**69.1**

GDRO is better than reweighting and subsampling for classifier retraining. Table 5 contains results on using different classifier retraining methods. We observe that GDRO produces the best group-robust performance (since GDRO is designed for this setting after all). Reweighting and subsampling seem to be effective on the vision datasets, but fail to perform on the NLP datasets.

Table 6: Comparison between retraining with GDRO on the full network (*Full*) versus last layer (*LL*).

GDRO $p = 0.30$	Waterbird		CelebA		MultiNLI		CivilComments	
	Avg Acc	Wg Acc	Avg Acc	Wg Acc	Avg Acc	Wg Acc	Avg Acc	Wg Acc
LL (CROIS)	91.4	**90.2**	91.6	**90.4**	80.3	**78.0**	89.7	69.1
Full	90.5	79.8	91.6	78.3	80.8	75.1	90.4	69.1

Classifier retraining outperforms full retraining with GDRO. Classifier retraining plays a central role in our method. In Table 6, we compare between fine-tuning with GDRO on the full DNN versus just the last layer (LL) for an independent split of $p = 0.30$. We see that GDRO on the LL is much better than full GDRO on most of the datasets (except CivilComments). However, the main difference is that while LL retraining (i.e. CROIS) requires little additional tuning, we must grid search for different regularization strength for GDRO when applied on the full network. CROIS can be shown to be quite robust to different parameter setting and regularization strength (details in Appendix C.1).

Models learned with ERM contain good features. The positive result for our decoupled training procedure provides another strong evidence for ERM trained models containing good features for the group-shift problem. While this is consistent with findings in the literatures on vision datasets (Kang et al., 2019; Menon et al., 2021), our work further provides some of the first evidences of this hypothesis in non-vision tasks, where the same result would not have been possible without independent split, as evident in the result for *Naive CRT* in Table 3.

Simplified model selection. The model selection criterion of picking the best average validation accuracy model simplifies hyperparameter tuning in comparison with other two-phases methods. This decision has been chosen mainly from the ablation experiments in Section B of the Appendix, where we observe that higher average validation accuracy generally suggests better features.

5 Conclusion and Future Works

In this paper, we propose Classifier Retraining on Independent Split (CROIS) as a simple method to reduce the amount of group annotations needed for improving worst-group performance as well as alleviate GDRO's requirement for careful control of model capacity. Our experimental results show the effectiveness of CROIS on four standard datasets across two settings and further provide evidences that ERM trained models contain good features for the group-shift problem.

Future works. The richness of ERM trained DNNs' features can potentially be useful towards solving the seemingly harder group-agnostic setting (where no group label is available) by allowing the practitioner to focus on obtaining a robust classifier given an ERM trained feature extractor, where we have shown that reasonable robustness can be achieved with relatively few group-labels, which makes the problem seem closer in reach. On a broader note, while most works in representation learning focus on producing good features (either with supervised, unsupervised, or self-supervised approaches), further examinations into different ways to perform classifier retraining in different settings (as in our work) could give a fuller picture to the features quality of different methods.

Reproducibility statement. We include our source code in the supplemental material. All implementation details and hyperparameters are detailed in Section A of the Appendix.

REFERENCES

M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

A. Ben-Tal, D. Den Hertog, A. De Waegenaere, B. Melenberg, and G. Rennen. Robust solutions of optimization problems affected by uncertain probabilities. *Management Science*, 59(2):341–357, 2013.

L. Biewald. Experiment tracking with weights and biases, 2020. URL `https://www.wandb.com/`. Software available from wandb.com.

D. Borkan, L. Dixon, J. Sorensen, N. Thain, and L. Vasserman. Nuanced metrics for measuring unintended bias with real data for text classification. In *Companion proceedings of the 2019 world wide web conference*, pages 491–500, 2019.

K. Cao, C. Wei, A. Gaidon, N. Arechiga, and T. Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in Neural Information Processing Systems*, 32:1567–1578, 2019.

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

N. Dagaev, B. D. Roads, X. Luo, D. N. Barry, K. R. Patil, and B. C. Love. A too-good-to-be-true prior to reduce shortcut reliance. *arXiv preprint arXiv:2102.06406*, 2021.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

J. C. Duchi, P. W. Glynn, and H. Namkoong. Statistics of robust optimization: A generalized empirical likelihood approach. *Mathematics of Operations Research*, 2021.

V. Feldman and C. Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. In *NeurIPS*, 2020.

A. Filos, P. Tigkas, R. McAllister, N. Rhinehart, S. Levine, and Y. Gal. Can autonomous vehicles identify, recover from, and adapt to distribution shifts? In *International Conference on Machine Learning*, pages 3145–3153. PMLR, 2020.

R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

K. Goel, A. Gu, Y. Li, and C. Re. Model patching: Closing the subgroup performance gap with data augmentation. In *International Conference on Learning Representations*, 2020.

I. Gulrajani and D. Lopez-Paz. In search of lost domain generalization. In *ICLR*. OpenReview.net, 2021.

G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing. Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications*, 73: 220–239, 2017.

T. Hashimoto, M. Srivastava, H. Namkoong, and P. Liang. Fairness without demographics in repeated loss minimization. In *International Conference on Machine Learning*, pages 1929–1938. PMLR, 2018.

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

B. Kang, S. Xie, M. Rohrbach, Z. Yan, A. Gordo, J. Feng, and Y. Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *International Conference on Learning Representations*, 2019.

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

D. Krueger, E. Caballero, J.-H. Jacobsen, A. Zhang, J. Binas, D. Zhang, R. Le Priol, and A. Courville. Out-of-distribution generalization via risk extrapolation (rex). In *International Conference on Machine Learning*, pages 5815–5826. PMLR, 2021.

P. Lahoti, A. Beutel, J. Chen, K. Lee, F. Prost, N. Thain, X. Wang, and E. Chi. Fairness without demographics through adversarially reweighted learning. *Advances in neural information processing systems*, 33:728–740, 2020.

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

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

A. K. Menon, A. S. Rawat, and S. Kumar. Overparameterisation and worst-case generalisation: friend or foe? In *International Conference on Learning Representations*, 2021.

J. Nam, H. Cha, S. Ahn, J. Lee, and J. Shin. Learning from failure: Training debiased classifier from biased classifier. *arXiv preprint arXiv:2007.02561*, 2020.

J. Nam, J. Kim, J. Lee, and J. Shin. Spread spurious attribute: Improving worst-group accuracy with spurious attribute estimation. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=_F9xpOrqyX9`.

L. Oakden-Rayner, J. Dunnmon, G. Carneiro, and C. Ré. Hidden stratification causes clinically meaningful failures in machine learning for medical imaging. In *Proceedings of the ACM conference on health, inference, and learning*, pages 151–159, 2020.

E. Rosenfeld, P. Ravikumar, and A. Risteski. An online learning approach to interpolation and extrapolation in domain generalization. *arXiv preprint arXiv:2102.13128*, 2021.

S. Sagawa, P. W. Koh, T. B. Hashimoto, and P. Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *International Conference of Learning Representation*, 2020a.

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

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

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

P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

A. Williams, N. Nangia, and S. R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*, 2017.

H. Yao, Y. Wang, S. Li, L. Zhang, W. Liang, J. Zou, and C. Finn. Improving out-of-distribution robustness via selective augmentation. *arXiv preprint arXiv:2201.00299*, 2022.

C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. In *ICLR*. OpenReview.net, 2017.

Y. Zhang, B. Kang, B. Hooi, S. Yan, and J. Feng. Deep long-tailed learning: A survey. *arXiv preprint arXiv:2110.04596*, 2021.

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

Table 7: Hyperparameters used in the experiments. The slash indicates the parameters used in the first phase (feature extractor) versus the second phase (classifier retraining).

	Waterbird	CelebA	MultiNLI	CivilComments
Learning Rate	$10^{-4}/10^{-4}$	$10^{-4}/10^{-4}$	$2 \times 10^{-5}/2 \times 10^{-5}$	$10^{-5}/10^{-5}$
ℓ_2 Regularization	$10^{-4}/10^{-4}$	$10^{-4}/10^{-4}$	$0/0$	$10^{-2}/0$
Number of Epochs	250/250	20/20	20/20	6/6

A Experimental Details

A.1 Infrastructure

We performed our experiment on 2 PCs with one NVIDIA RTX3070 and one NVIDIA RTX3090. Our implementation is built on top of the code base from Liu et al. (2021). Experimental data was collected with the help of Weights and Biases (Biewald, 2020).

A.2 Hyperparameters

Table 7 contains the hyperparameters used in our experiments in Sections 4.2 and 4.1. Note that these are the standard parameters for obtaining an ERM model for these datasets as in previous works (Sagawa et al., 2020a; Liu et al., 2021). The only difference is that we train Waterbird and CelebA for slightly shorter epoch due to finding no further increase in validation accuracies after those epochs.

In our experiments, unless noted, we do not tune for any other hyperparameters. For the second phase of CivilComments, we do not use the default regularization but opt for 0 since the linear layer already has low capacity. However, adding further regularization does not seem to have much of an effect as in section C.1.

B Ablation studies: Obtaining a good feature extractor

B.1 Impact of The Feature Extractor's Algorithms

Here, we provide evidences that ERM trained models produce the best features for worst-group robustness. We conduct an experiment on Waterbird, where instead of using ERM to obtain a feature extractor, we perform GDRO and Reweighing instead on the first phase. The results are in table 8. While using reweighing or GDRO for the first phase defeats the purpose of reducing the amount of group-labels needed (whereas ERM doesn't need any), it is informative to examine the features alone.

Table 8: Effects of different methods for obtaining a feature extractor on test average accuracy and test worst-group accuracy (with ResNet50 on Waterbird).

Feature extractor via	Test Avg Acc	Test Wg Acc
Reweighing	90.1	88.8
GDRO	90.8	88.6
ERM	90.5	**90.2**

Here, we see that even though ERM does not use group-labels, it provides the best features for robust classifier retraining on independent split.

B.2 Impact of Early Stopping and Validation Accuracies on The Feature Extractor

In this section, we present an ablation study of how different early stopping epoch (figure 4), average validation accuracy (figure 5 left), and worst-group accuracy (figure 5 right) of the initial ERM trained model (feature extractor) affect the GDRO retraining phase of CROIS. The results here are from

performing CROIS with GDRO and $p = 0.30$ on Waterbird across a wide variety of epochs. Table 9 presents the full data generated for this section.

Figure 4: The effect of using different epochs for the feature extractor (phase 1) on classifier retraining's (phase 2) test accuracies.

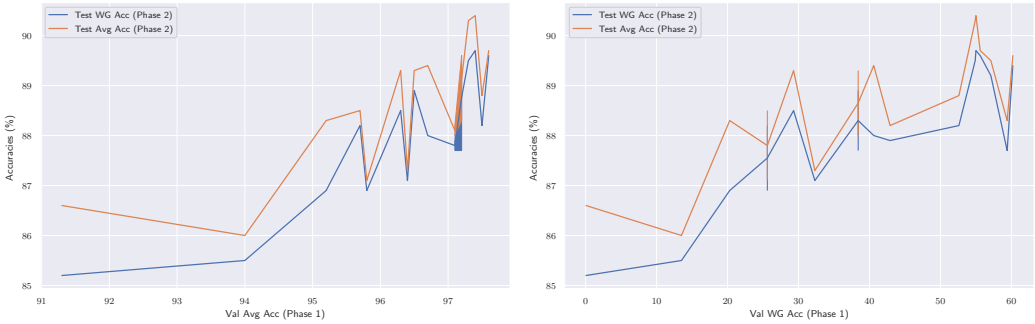

Figure 5: The effect of different *validation average accuracies* (**left**) and *validation worst-group accuracies* (**right**) from the feature extractor (phase 1) on classifier retraining's (phase 2) test accuracies.

C HYPERPARAMETER TUNING: CROIS VS. GDRO

C.1 FURTHER HYPERAMETER EXPLORATION ON CROIS

In our paper, we have demonstrated CROIS's effectiveness even with just using the same hyperparameters as to train an ERM model. In this section, we present results for further additional parameter tuning on the robust classifier retraining phase. These results provide empirical evidence for CROIS's potential as well as robustness to different hyperparameter setting.

ℓ_2 regularization. We investigate whether *additional* regularization would be helpful to classifier retraining with GDRO on CelebA (Table 10) and Waterbird. (Table 11) We further examine the effects of regularization on CivilComments (Table 12) to support our choice in Section A. For these experiments, we further run each setting acorss 3 random seeds to further investigate CROIS's sensitivity towards hyperparameters.

Table 9: CROIS with GDRO ($p = 0.30$) on Waterbird. Average (*Avg Acc*) and Worst-group (*Wg Acc*) Accuracies for various epochs of the feature extractor ("Phase 1") and the corresponding test accuracies for classifier retraining ("Phase 2"). While training for longer epochs seem to help with average and worst-group accuracy for phase 2, the benefit is small. Hence, simply selecting the best validation average accuracy model, row Epoch 131 and denoted *BEST* here, yields good enough features that simplify our training procedure and model selection criteria.

CROIS ($p = 0.3$)	Feature Extractor (Phase 1)		Classifier Retraining (Phase 2)			
Phase 1 Epoch	Val Avg	Val WG	Val Avg	Val WG	Test Avg	Test WG
0	91.3	0.05	87.8	85.6	86.6	85.2
1	94	13.5	88	87.6	86	85.5
2	95.2	20.3	89.6	88	88.3	86.9
3	95.7	25.6	90.1	88	88.5	88.2
4	95.8	25.6	88.9	88.2	87.1	86.9
5	96.4	32.3	89.2	88.7	87.3	87.1
6	96.5	38.4	90.5	88.7	89.3	88.9
7	96.3	29.3	90.6	88.7	89.3	88.5
8	97.1	38.4	90.1	89.3	88	87.7
9	97.1	42.9	90.4	89.9	88.2	87.9
10	96.7	40.6	91.2	90.2	89.4	88
20	97.3	54.9	91.4	91	90.3	89.5
50	97.4	55	**91.5**	91	**90.4**	**89.7**
100	97.2	57.1	90.5	90.2	89.5	89.2
131 (Best)	**97.6**	55.6	90.9	90.2	89.7	**89.6**
150	97.2	**60.2**	90.7	90.2	89.6	89.4
200	97.5	52.6	91	90.2	88.8	88.2
250	97.2	59.4	91	**90.4**	88.3	87.7

Table 10: Effects of ℓ_2 regularization on classifier retraining with GDRO on CelebA.

	$p = 0.50$		$p = 0.30$		$p = 0.10$	
ℓ_2 Reg.	Avg Acc	Wg Acc	Avg Acc	Wg Acc	Avg Acc	Wg Acc
1	91.6 (0.06)	87.4 (2.76)	90.9 (1.13)	88.4 (0.78)	91.2 (0.35)	90.0 (0.70)
10^{-2}	92.1 (0.32)	87.6 (1.56)	**91.8** (0.21)	87.5 (3.54)	92.0 (0.44)	88.3 (2.39)
10^{-4}	91.9 (0.35)	**88.2** (2.10)	91.3 (0.44)	**90.6** (0.95)	91.3 (0.36)	**90.3** (0.82)
0	**92.2** (0.25)	86.8 (2.30)	91.5 (0.07)	87.8 (3.96)	**92.1** (0.47)	88.1 (2.73)

Table 11: Effects of ℓ_2 regularization on classifier retraining with GDRO on Waterbird.

	$p = 0.50$		$p = 0.30$		$p = 0.10$	
ℓ_2 Reg.	Avg Acc	Wg Acc	Avg Acc	Wg Acc	Avg Acc	Wg Acc
1	89.6 (0.93)	88.9 (1.37)	**91.7** (0.44)	**89.8** (0.68)	94.5 (0.50)	85.8 (0.15)
10^{-2}	**90.4** (1.31)	89.3 (0.61)	90.6 (1.25)	89.2 (1.31)	95.1 (0.40)	86.3 (0.83)
10^{-4}	**90.4** (0.95)	**89.5** (0.59)	90.8 (0.35)	89.6 (1.15)	**95.4** (1.10)	83.5 (3.24)
0	89.8 (0.53)	89.3 (0.70)	90.6 (1.31)	89.1 (1.16)	95.1 (0.35)	**86.4** (0.90)

Table 12: Effects of ℓ_2 regularization on classifier retraining with GDRO on CivilComments.

	$p = 0.50$		$p = 0.30$		$p = 0.10$	
ℓ_2 Reg.	Avg Acc	Wg Acc	Avg Acc	Wg Acc	Avg Acc	Wg Acc
1	88.8 (1.30)	70.0 (1.63)	89.5 (0.35)	66.4 (1.99)	89.3 (1.69)	68.9 (2.64)
10^{-2}	89.4 (0.99)	70.6 (0.42)	89.5 (0.29)	68.5 (0.87)	88.6 (1.70)	**70.2** (1.63)
0	**89.5** (0.70)	**71.0** (1.50)	**89.7** (0.33)	**68.6** (1.53)	**89.5** (1.81)	68.7 (1.72)

Learning rate. We examine the effects of different learning rates on CROIS on CelebA and Waterbird in Table 13. Lower learning rate seems to be more beneficial.

Table 13: Effects of varying learning rate on CROIS $p = 0.30$ on CelebA and Waterbird. We fix ℓ_2 regularization to 10^{-4}.

	Learning rate	10^{-5}	10^{-4}	10^{-3}	10^{-2}	10^{-1}
CelebA	Average accuracy	**92.2**	91.4	91.2	90.1	91.2
	Worst-group accuracy	88.3	90	**90.3**	87.8	82.8
Waterbird	Average accuracy	89.7	90.6	**94.2**	93.2	93.9
	Worst-group accuracy	**89.5**	88.9	87.1	88.8	78.5

C.2 Sensitivity to model capacity: CROIS versus GDRO

In this section, we present a comparison between CROIS and GDRO test performance with different ℓ_2 regularization configuration on CelebA and Waterbird in Table 4. On CelebA, GDRO is quite sensitive to model capacity while it is less so on Waterbird. We note that GDRO fails to converge to a good stationary point when $\ell_2 = 1$ on CelebA (see Figure 3).

D Further studies on classifier retraining

D.1 Impact of Robust Retraining on Independent Splits

In this section, we examine how robust retraining affects the model's prediction of D_U and D_L before and after robust classifier retraining on independent split (with $p = 0.30$). Tables 14 and 15 show the accuracy on D_U and D_L for Waterbird and CelebA. The "Points changed" column indicates the number of points that the model changes prediction after robust retraining per group along with the total number of examples in that group (with percentage in parentheses). The worst-group is underlined in the tables.

Table 14: Model's prediction of D_U and D_L on Waterbird *before* robust retraining and *after* robust classifier retraining on independent split.

Waterbird ($p = 0.3$)	Accuracy on D_U			Accuracy on D_L		
	Before	After	Points changed	Before	After	Points changed
Avg Acc	100	94.5	184/3356 (5.48%)	96.4	89.6	162/1439 (11.3%)
Group 0 (73.0%)	100	<u>92.6</u>	180/2430 (7.41%)	99.6	<u>88.2</u>	122/1068 (11.4%)
Group 1 (3.84%)	100	99.3	1/141 (7.09%)	76.7	100	10/43 (23.3%)
Group 2 (1.17%)	100	100	0/38 (0.00%)	<u>44.4</u>	100	10/18 (55.6%)
Group 3 (22.0%)	100	99.6	3/747 (0.40%)	90.8	92.3	20/310 (6.45%)

Table 15: Model's prediction of D_U and D_L on CelebA before and after robust classifier retraining on independent split. RR means after Robust Retraining.

CelebA ($p = 0.30$)	Accuracy on D_U			Accuracy on D_L		
	Before	After	Points changed	Before	After	Points changed
Avg Acc	96.5	92.4	7537/113939 (6.61%)	95.6	92.0	3274/48831 (11.3%)
Group 0 (43.7%)	96.7	91.7	2609/50311 (5.19%)	95.8	91.3	1011/21318 (4.74%)
Group 1 (41.4%)	99.6	92.2	3489/46652 (7.48%)	99.6	92.2	1493/20222 (7.38%)
Group 2 (14.1%)	89.9	95.2	1000/16012 (6.25%)	86.8	93.7	517/6868 (7.53%)
Group 3 (0.87%)	46.3	91.8	439/964 (45.4%)	35.5	95.3	253/423 (59.8%)

Interestingly, after robust retraining, the worst-group almost always switches from the minority group to the majority group regardless of the data split.

D.2 The Importance of Minority Examples: Subsampled Retraining

As alluded to in Section 4.2, group-imbalance seems to play an important role in the robust performance of CROIS. To further demonstrate this point, we consider how CROIS performs when the second phase is subsampled versus when it is allowed additional non-minority examples.

Table 16: Performance of CROIS when retraining is on a subsampled split versus the full split. Subsampling the split doesn't seem to impact CROIS's performance, indicating the importance of the availability of minority examples.

Dataset (minority group size, fraction)	Subsampled retraining		Full retraining	
	Avg Acc	Wg Acc	Avg Acc	Wg Acc
CelebA ($n = 423$, 0.87%)	90.6	89.7	91.9	88.9
Waterbird ($n = 18$, 1.2%)	91.3	87.6	90.3	88.9

As the result in Table 16 shows, there isn't a significant difference between the two sampling strategies, suggesting that the availability of minority group examples plays an important role in robust classifier retraining.

E Fraction of the validation set implementation details from Section 4.1

Following the setup in Section 4.1 and the setup as in Liu et al. (2021); Nam et al. (2022), we further reduce the validation set to only a small fraction, 5%, 10%, and 20%. We investigate CROIS's performance in this very few group-labels setting across CelebA and Waterbird in Section 4.1.

We note that the highly reduced sample size poses new challenge and makes it harder to simply reuse the default parameters. Hence, we tune the learning rate (across $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$), ℓ_2 regularization (across $\{0, 10^{-4}, 10^{-2}\}$) and the batch size (details below) for CROIS to work with the smaller amount of examples. We also perform model selection via the training accuracy instead of an independently split.

- **The use of group-labels and model selection:** Since the amount of examples for classifier retraining is now significantly reduced, it might be wasteful to further split our available group-labels for validation. Instead, we use all the available group-labels for robust classifier retraining and perform model selection in the second phase via the *train worst-group accuracy*. The low capacity linear layer allows us to avoid overfitting when performing model selection this way. The feature extractor from the first phase is selected via the best average accuracy on the full group-unlabeled validation set.

- **Smaller batch size:** Since GDRO requires group-balanced sampling, a batch size greater than the number of examples in a certain group would cause duplicate sampling of the minority-group examples in the same step, artificially increasing the weight for that group. We further tune for batch sizes across a grid of powers of 2 less than the smallest group or the default batch size (e.g. we search across $\{4, 8, 16\}$ if the size of the smallest group is 17).
- **Regularization:** When using so little data, overfitting can become a bigger problem, even when just training a low-capacity linear classifier. Hence, we tune for ℓ_2 regularization.

In Table 2, we present the results for CROIS with the above modifications and compare it to CROIS and JTT. There, robust retraining for CelebA is performed with an ℓ_2 regularization of 0.1 and Waterbird with group-adjustments equalling 1 (these are similar to Sagawa et al. (2020a)).

This result shows that CROIS maintains its robustness performance even at greatly reduced group-labels. This implies that even a few (minority) examples can help debias the final layer classifier with proper configurations.

F IMPLICATIONS AND TAKEAWAYS

Comparison to standard pretraining and finetuning As mentioned in the related works section as well as during the discussion of the motivation for our method, pretraining and then finetuning is a now well-known and established strategy in many domains. Our work differs the most significantly from this standard strategy through:

1. The use of independent splits: Traditional pretraining and finetuning reuses the dataset for both phase with the possibility of additional labels (contrastive learning, long-tail learning, etc.). We have demonstrated through extensive experiments the importance of independent split for classifier retraining to work well for the group robust setting.
2. The use of a group robust algorithm for finetuning: We mainly utilize GDRO for our finetuning phase while most other works in finetuning make use of simpler strategy for finetuning like reweighing or subsampling. We demonstrated in our experiments that GDRO yields the best robust performance over other more common methods.

Our work provides evidences that the features of ERM trained DNNs are rich enough to solve the group-shift problem (when an abundant amount of group labels is available to retrain the classifier) and one of the major reasons for poor worst-group performance of an ERM trained DNN is within its classifier layer. We then further demonstrate that even a few group labels can sufficiently "fix" the classifier to achieve better group-robust performance.

This knowledge can potentially be useful towards solving the seemingly much harder group-agnostic setting (where no group label is available) by allowing the practitioner to focus on obtaining a robust classifier given an ERM trained feature extractor. Our experiments further show that reasonable robustness can be achieved with relatively few group-labels (that are not used to obtain the feature extractor), which makes the problem seem closer in reach.