On Feature Learning in the Presence of Spurious Correlations

Pavel Izmailov* **Polina Kirichenko*** **Nate Gruver*** **Andrew Gordon Wilson**
New York University

Abstract

Deep classifiers are known to rely on spurious features — patterns which are correlated with the target on the training data but not inherently relevant to the learning problem, such as the image backgrounds when classifying the foregrounds. In this paper we evaluate the amount of information about the core (non-spurious) features that can be decoded from the representations learned by standard empirical risk minimization (ERM) and specialized group robustness training. Following recent work on Deep Feature Reweighting (DFR), we evaluate the feature representations by re-training the last layer of the model on a held-out set where the spurious correlation is broken. On multiple vision and NLP problems, we show that the features learned by simple ERM are highly competitive with the features learned by specialized group robustness methods targeted at reducing the effect of spurious correlations. Moreover, we show that the quality of learned feature representations is greatly affected by the design decisions beyond the training method, such as the model architecture and pre-training strategy. On the other hand, we find that strong regularization is not necessary for learning high-quality feature representations. Finally, using insights from our analysis, we significantly improve upon the best results reported in the literature on the popular Waterbirds, CelebA hair color prediction and WILDS-FMOW problems, achieving 97%, 92% and 50% worst-group accuracies, respectively.

1 Introduction

In classification problems, a feature is *spurious* if it is predictive of the label without being causally related to it. Models that exploit the predictive power of spurious features can achieve strong average performance on training and in-distribution test data but often perform poorly on sub-groups of the data where the spurious correlation does not hold [18]. For example, neural networks trained on ImageNet are known to rely on backgrounds [93] or texture [19], which are often correlated with labels without being causally significant. Similarly in natural language processing, models often rely on specific words and syntactic heuristics when predicting the sentiment of a sentence or the relationship between a pair of sentences [56, 22]. In extreme cases, neural networks completely ignore task-relevant *core* features and only use spurious features in their predictions [97, 79], achieving zero accuracy on the subgroups of the data where the spurious correlation does not hold.

In recent work, Kirichenko et al. [40] showed that, surprisingly, standard Empirical Risk Minimization (ERM) learns a high-quality representation of the core features on datasets with spurious correlations, even when the model primarily relies on spurious features to make predictions. Moreover, they showed that it is often possible to recover state-of-the-art performance on benchmark problems by simply retraining the last layer of the model on a small held-out dataset where the spurious correlation does not hold. This procedure is called Deep Feature Reweighting (DFR).

*Equal contribution.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

In this paper, we provide an in-depth study of the factors that affect the *quality of learned representations* in the presence of spurious correlations: how accurately can we decode the core features from the learned representations. Following Kirichenko et al. [40], we break the problem of training a robust classifier into two tasks: extracting feature representations and training a linear classifier on these features. In order to study the feature learning in isolation, we use the DFR procedure to learn an optimal linear classifier on the feature representations, and evaluate the features learned with different training methods, neural network architectures, and hyper-parameters.

First, on a range of problems with spurious correlations we show that while specialized group robustness methods such as group distributionally robust optimization (group DRO) [76] can significantly outperform the standard ERM training, the quality of the features learned by ERM is highly competitive: by applying the DFR procedure to the features learned by ERM and group DRO we achieve similar performance. Furthermore, we show that the performance improvements of group DRO are largely explained by the better weighting of the learned features in the last classification layer, and not by learning a better representation of the core features. This observation has high practical significance, as the problem of training the last layer of the model is much simpler both conceptually and computationally than training the full model to avoid spurious correlations [40].

Next, focusing on the ERM training, we explore the effect of model class, pretraining strategy and regularization on feature learning. We find a linear dependence between the in-distribution accuracy of the model and the worst group accuracy after applying DFR, meaning that on natural datasets good generalization typically implies good feature learning, even in the presence of spurious features. Further, we show that the pre-training strategy has a very significant effect on the quality of the learned features, while strong regularization does not significantly improve the feature representations on most benchmarks.

Finally, by finetuning a state-of-the-art ConvNext model [52], we significantly outperform the best reported results on the popular Waterbirds [76], CelebA hair color and WILDS FMOW [43] spurious correlation benchmarks, using only simple ERM training followed by DFR.

Our code is available at `github.com/izmailovpavel/spurious_feature_learning`.

2 Related Work

Numerous works describe how neural networks can rely on spurious correlations in real world problems. In vision, neural networks can learn to rely on an image's background [93, 76, 61], secondary objects [45, 72, 83, 80, 2, 60], object textures [19] and other semantically irrelevant features [9, 49]. Spurious correlations are especially problematic in high-risk domains such as medical imaging, where it was shown that neural networks can use hospital-specific metal tokens [97] or cues of disease treatment [65] rather than symptoms to perform automated diagnosis on chest X-ray images. Spurious features are also extremely prevalent in NLP, where models can achieve good performance on benchmarks without properly solving them, e.g. by using simple syntactic heuristics such as lexical overlap between the two sentences in order to classify the relationship between them [64, 22, 37, 56]. For a comprehensive survey of the area, see Geirhos et al. [19].

Because of the high practical significance of spurious correlations, many *group robustness* methods have been proposed. These methods aim to reduce the reliance of deep learning models on spurious correlations and improve worst group performance. Group DRO [76] is the state-of-the-art group robustness method, which minimizes the worst-group loss instead of the average loss. Other works focus on automatically identifying the minority group examples [50, 62, 13, 100], learning several diverse classifiers that use different features [48, 66, 85] or using partially available group labels [84, 63]. Group subsampling was shown to be a strong baseline for some benchmarks [34, 78].

In this work, we focus on *feature learning* in the presence of spurious correlations. Hermann and Lampinen [31] perform a conceptually similar study, but focusing on synthetic datasets. Similar to their work, we explore how well the different features of the data can be decoded from the features learned by deep neural networks, but on large-scale natural datasets. Hermann et al. [30] explore the feature learning in the context of texture bias [19], finding that data augmentation has a profound effect on the texture bias while architectures and training objectives have a relatively small effect. Ghosal et al. [20] show that Vision Transformer models pretrained on ImageNet22k [44] significantly outperform standard CNN models on several spurious correlation benchmarks.

Lovering et al. [54] explore the factors which affect the extractability of features after pre-training and fine-tuning of NLP models. Kaushik et al. [36] construct counterfactually augmented sentiment analysis and naural language inference datasets (CAD) and show that combining CAD with the original data reduces the reliance on spurious correlations on the corresponding benchmarks. Kaushik et al. [38] explain the efficacy of CAD and show that while adding noise to causal features degrades in-distribution and out-of-distribution performance, adding noise to non-causal features improves robustess. Eisenstein [17] and Veitch et al. [86] formally define and study spurious features in NLP from the perspective of causality.

Kirichenko et al. [40] show that models trained with standard ERM training often learn high-quality representations of the core features, and propose the DFR procedure (see Section 3) which we use extensively in this paper. Related observations have also been reported in other works in the context of spurious correlations [58], domain generalization [73] and long-tail classification [35]. While we build on the observations of Kirichenko et al. [40], our work provides profound new insights and greatly expands on the scope of their work. In particular, we investigate the feature representations learned by methods beyond standard ERM, and the role of model architecture, pre-training, regularization and data augmentation on learning semantic structure. We also extend our analysis beyond the standard spurious correlation benchmarks studied by Kirichenko et al. [40], by considering the challenging real world satellite imaging and chest X-ray datasets.

In an independent and concurrent work, Shi et al. [81] also propose an evaluation framework for out-of-distribution generalization based on last layer retraining, inspired by the observations of Kirichenko et al. [40] and Kang et al. [35]. They focus on the comparison of supervised, self-supervised and unsupervised training methods, providing complementary observations to our work.

3 Background

Preliminaries. We consider classification tasks with inputs $x \in \mathcal{X}$ and classes $y \in \mathcal{Y}$. We assume that the data distribution consists of groups $\mathcal{G}$ which are not equally represented in the training data. The distribution of groups can change between the training and test distributions, with majority groups becoming less common or minority groups becoming more common. Because of the imbalance in training data, models trained with ERM often have a gap between average and worst group performance on test. Throughout this paper, we will be studying *worst group accuracy* (*WGA*), i.e. the lowest test accuracy across all the groups $\mathcal{G}$. For most problems considered in this paper, we assume that each data point has an attribute $s \in \mathcal{S}$ which is spuriously correlated with the label y, and the groups are defined by a combination of the label and spurious attribute: $\mathcal{G} \in \mathcal{Y} \times \mathcal{S}$. In test distribution we might find that s is no longer correlated with y, and thus a model that has learned to rely on the spurious feature s during training will perform poorly at test time. Models that rely on the spurious features will typically achieve poor worst group accuracy, while models that rely on core features will have more uniform accuracies across the groups. In Appendix A, we describe the groups, spurious and core features in the datasets that we use in this paper.

In order to perform controlled experiments, we assume that we have access to the spurious attributes s (or group labels) for training or validation data, which we use for training of group robustness baselines and feature quality evaluation. However, we emphasize that our results on the features learned by ERM hold generally, even when spurious attributes are unknown, as ERM does not use the information about the spurious features: we only use the spurious attributes to perform analysis.

Deep feature reweighting. Suppose we are given a model $m : \mathcal{X} \to \mathcal{C}$, where $\mathcal{X}$ is the input space and $\mathcal{C}$ is the set of classes. Kirichenko et al. [40] assume that the model m consists of a feature extractor (typically, a sequence of convolutional or transformer layers) followed by a classification head (typically, a single linear layer): $m = h \circ e$, where $e : \mathcal{X} \to \mathcal{F}$ is a feature extractor and $h : \mathcal{F} \to \mathcal{C}$ is a classification head. They discard the classification head, and use the feature extractor e to compute the set of embeddings $\hat{\mathcal{D}}_e = \{(e(x_i), y_i)\}_{i=1}^n$ of all the datapoints in the reweighting dataset $\hat{\mathcal{D}}$; the reweighting dataset is used to retrain the last layer of the model, and contains group-balanced data where the spurious correlation does not hold. Finally, they train a logistic regression classifier $l : \mathcal{F} \to \mathcal{C}$ on the dataset $\hat{\mathcal{D}}_e$. For stability, logistic regression models are trained 10 times on different random group-balanced subsets of the reweighting dataset $\hat{\mathcal{D}}$, and the weights of the learned logistic regression models are averaged (see Appendix B of Kirichenko et al. [40] for full details on the DFR procedure). Then, the final model used on new test data is given by $m_l = l \circ e$.

Thoughout this paper, we use a group-balanced held-out dataset (subset of the validation dataset where each group has the same number of datapoints) as the reweighting dataset $\hat{\mathcal{D}}$; Kirichenko et al. [40] denote this variation of the method as $\text{DFR}^{\text{Val}}_{\text{Tr}}$.

4 Experimental Setup and Evaluation Procedure

In this section, we describe the datasets, models and evaluation procedure that we use throughout the paper.

Datasets. In order to cover a broad range of practical scenarios, we consider four image classification and two text classification problems.

- *Waterbirds* [76] is a binary image classification problem, where the class corresponds to the type of the bird (landbird or waterbird), and the background is spuriously correlated with the class. Namely, most landbirds are shown on land, and most waterbirds are shown over water.
- *CelebA hair color* [51] is a binary image classification problem, where the goal is to predict whether a person shown in the image is blond; the gender of the person serves as a spurious feature, as 94% of the images with the "blond" label depict females.
- WILDS-*FMOW* [12, 43, 77] is a satellite image classification problem, where the classes correspond to one of 62 land use or building types, and the spurious attribute s corresponds to the region (Africa, Americas, Asia, Europe, Oceania or Other; the "Other" region is not used in the evaluation). We note that for the FMOW datasets the groups $\mathcal{G}_s$ are defined by the value of the spurious attribute, and not the combination of the spurious attribute and the class label $\mathcal{G}_{y,s}$, as described in Section 3. Moreover, on FMOW there is also a domain shift: the images for test and validation data (used for last layer retraining) are collected in 2016 and 2017, while the training data is collected before 2016. For more details, please see Appendix A.
- *CXR-14* [89] is a dataset with chest X-ray images for which we focus on a binary classification problem of pneumothorax prediction. Oakden-Rayner et al. [65] showed that there is a hidden stratification in the dataset such that most images from the positive class contain a chest drain, which is a non-causal feature related to treatment of the disease. While for all other benchmarks we report WGA on test data, for this dataset, following prior work [65, 48, 70], we report worst group AUC because of the heavy class imbalance.[1]
- *MultiNLI* [91, 76] is a text classification problem, where the task is to classify the relationship between a given pair of sentences as a contradiction, entailment or neither of them. In this dataset, the presence of negation words (e.g. "never") in the second sentence is spuriously correlated with the "contradiction" class.
- *CivilComments* [8, 43] is a text classification problem, where the goal is to classify whether a given comment is toxic. We follow Idrissi et al. [34] and use the coarse version of the dataset both for training and evaluation, where the spurious attribute is $s = 1$ if the comment mentions at least one of the following categories: male, female, LGBT, black, white, Christian, Muslim, other religion; otherwise, the spurious label is 0. The presence of the eight categories above is spuriously correlated with the comment being classified as toxic.

The Waterbirds, CelebA, CivilComments and MultiNLI datasets are commonly used to benchmark the performance of group robustness methods [see e.g. 34, 50, 63]. The FMOW and CXR-14 datasets present challenging real-world problems with spurious correlations. In these datasets, the inputs do not resemble natural images from datasets such as ImageNet [75], so models have to learn the relevant features from data to achieve good performance, and cannot simply rely on feature transfer. We provide detailed descriptions of the data and show example datapoints in Appendix A, Figures 6 and 7.

[1] We take the minimum out of the two scores on test: AUC for classifying negative class against positive examples with chest drain and AUC for negative class against positive without chest drain.

Models. Following prior work [76, 34] we use a ResNet-50 [26] model pretrained on ImageNet1k [75] on Waterbirds, CelebA and FMOW. For the NLP problems, we use a BERT model [14] pretrained on Book Corpus and English Wikipedia data. On CXR-14, following prior work [e.g. 70, 65, 48] we use a DenseNet-121 model [32] pretrained on ImageNet1k. In Section 6, we provide an extensive study of the effect of architecture and pretraining on the image classification problems, and in Appendix E we perform a similar study on the MultiNLI text classification problem.

Evaluation strategy. We use DFR to evaluate the quality of the learned feature representations, as described in Section 3: we measure how well the core features can be decoded from the learned representations with last layer retraining. In some of the experiments, we also train a classifier to predict the spurious attribute s instead of the class label y. Using this classifier, we can evaluate the decodability of the spurious feature from the learned feature representation. We refer to this procedure as s-DFR and the corresponding worst group accuracy (in predicting the spurious attribute s) as DFR s-WGA. Additionally, we evaluate the worst group accuracy and mean[2] accuracy of the base model without applying DFR, which we refer to as *base WGA* and *base accuracy* respectively.

5 ERM vs Group Robustness Training

Multiple methods have been proposed for training classifiers which are more robust to spurious correlations, with significant improvements in worst group accuracy compared to standard training. In this section, we use DFR to investigate whether the improvements of group robustness methods are caused by better feature representations or by better weighting of the learned features.

Methods. We consider 4 methods for learning the features. *ERM* or Empirical Risk Minimization is the standard training on the original training data, without any techniques targeted at improving worst group performance. *RWG* reweights the loss on each of the groups according to the size of the group and *RWY* reweights the loss on each class according to the size of the class [34]. *Group DRO* [76] is a state-of-the-art method which uses the group information on the training data to minimize the worst group loss instead of the average loss. Group DRO is often considered as an oracle method or upper-bound on the worst group performance under spurious correlations [50, 13].

On the CXR dataset the group labels are not available on the train data, so we cannot apply RWG or group DRO; on this dataset we compare ERM to RWY. On several datasets, the performance of RWY and RWG methods deteriorates during training. For these datasets, we additionally report the results for the checkpoint obtained with early stopping (RWY-ES and RWG-ES). For group DRO, we report the performance with early stopping on all datasets except for CXR (GDRO-ES). In all cases, early stopping is performed based on the worst-group accuracy on the validation set.

Hyper-parameters selection. We train ERM models, RWG and RWY with the same hyper-parameters shared between all the image datasets (apart from batch size which is set to 32 on Waterbirds, and 100 on the other datasets) and between the natural language datasets. We do not tune the hyper-parameters of these methods for worst group accuracy. For group DRO, we run a grid search over the values of the generalization adjustment C, weight decay and learning rate hyper-parameters, and select the best combination according to the worst-group accuracy on validation data with early stopping. For details, please see Appendix B.

5.1 Results

We compare feature learning methods on all datasets in Figure 1. As expected, the worst group accuracy of group robustness methods is significantly better than the ERM worst group accuracy on most datasets. For example, on Waterbirds, ERM only gets 68.8% WGA, while group DRO with early stopping gets 90.6%. However, after applying DFR the performance of ERM and group DRO is very close, with a slight advantage for ERM (91.1% for ERM and 90% for group DRO), and similar observations hold on all datasets.

The results for RWG and RWY are analogous. Namely, when combined with early stopping, these methods outperform ERM on base model performance. Once we apply DFR, however, the gap in

[2]Following Sagawa et al. [76] and Kirichenko et al. [40], we evaluate the mean accuracy according to the group distribution in the training set. This way, mean accuracy represents the in-distribution generalization of the model.

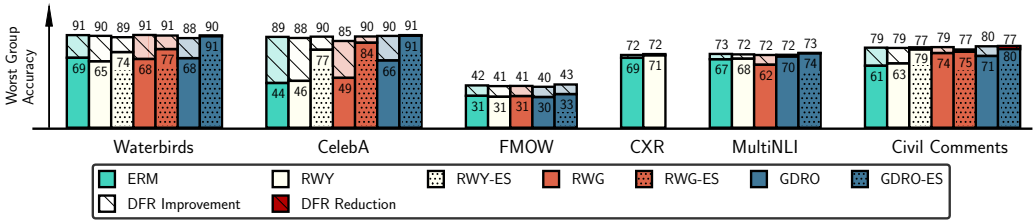


Figure 1: **ERM vs group robustness methods.** Performance of ERM on group robustness methods on vision and NLP benchmark problems. *ES* stands for early stopping. For each method on each dataset we report the base model worst group accuracy (shown with the number inside each bar), and the worst group accuracy after applying DFR (shown above each bar). On CXR-14, we report worst group AUC. While on many datasets the base performance of ERM is much worse compared to group robustness methods, the performance of the different methods is similar after we apply DFR, suggesting that the strong performance of group robustness methods is largely caused by better weighting of the learned features rather than better feature representations.

performance between the methods becomes very small. In fact, on all the datasets and for all the methods the improvement in worst group accuracy from using any of the considered group robustness methods compared to ERM does not exceed 1–2% after applying DFR.

These results suggest that the improvements over ERM in base model performance for methods such as group DRO and RWG are largely the result of better weighting of the learned features rather than learning better representations of the core features. Indeed, if the core feature was better represented by group robustness methods, we would expect to see a significant improvement over ERM after applying DFR.

This observation is significant both practically and scientifically. Practically, the problem of training the last layer is simpler, more data efficient and less compute intensive than retraining the full model [40]. Our results suggest that for many problems, practitioners can primarily focus on retraining the last layer, as training the feature extractor model with group robustness methods does not provide significant improvements. Scientifically, robustness to spurious correlations is often implicitly or explicitly associated with the quality of learned feature representations [e.g. 3, 4, 74, 100, 48, 99]. Our results suggest that the quality of feature representations is not significantly affected by group DRO, refining our understanding of group robustness training and representation learning in the presence of spurious correlations.

Effect of early stopping. Early stopping is crucial to achieving strong base model performance with RWY, RWG and group DRO on many of the datasets. In Figure 1 we report the results both with and without early stopping on datasets where the validation worst group accuracy significantly degrades over the course of training. Generally, early stopping does not appear to significantly improve the quality of the learned feature representations even in these problems: after applying DFR, methods with and without early stopping achieve similar performance. This observation suggests that late in training, neural networks may start to assign a higher weight to the spurious features, but the information about the core features is still preserved in the learned representations. For ERM and Group DRO, we explore the DFR WGA performance as a function of the training iteration in Appendix D, also finding that the length of training has a relatively small effect on the final DFR WGA.

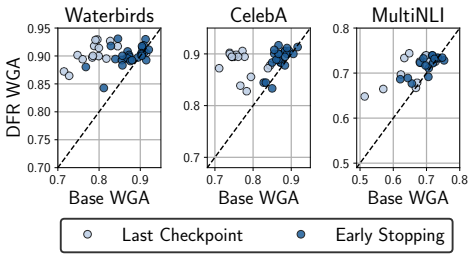


Figure 2: **Group DRO.** Worst group accuracy before and after applying DFR for a range of group DRO runs. DFR does not improve the best runs, indicating that group DRO already learns a nearly optimal last layer.

Group DRO analysis. In Figure 2 we report the worst group accuracy of multiple group DRO runs before and after applying DFR. For each run, we evaluate the best checkpoint according to validation

accuracy (i.e. the checkpoint selected by early stopping) and the last checkpoint saved after a fixed number of epochs. We observe that while the last checkpoints perform poorly in terms of base WGA in most runs, DFR can significantly improve their performance, removing the need for early stopping. Interestingly, we find that the best performing group DRO models cannot be improved by DFR on each of the datasets. This result suggests that the weighting of the features learned by group DRO is already close to optimal, again indicating that the success of group DRO can largely be attributed to learning a better weighting for the features in the last linear layer, rather than learning better features.

Group robustness methods (e.g. group DRO) perform well because they improve the last linear layer, not the underlying feature representations.

6 Effect of the Base Model

Most of the prior work on spurious correlation considers a fixed model class for each problem: for example, on Waterbirds and CelebA datasets almost all the papers use a ResNet-50 base model pre-trained on ImageNet1k [e.g. 76, 50, 48, 40, 84, 63, 100]. Recently, Ghosal et al. [20] showed that vision transformer models [15] may provide better robustness to spurious correlations if pretrained on a large dataset. Here, we perform a systematic large-scale evaluation of the effect of base model choice on the quality of learned feature representations.

We repeat the experiments on the effect of base model, pretraining, and training on the target dataset presented in this section on the MultiNLI text classification task in Appendix E, with similar observations.

6.1 Effect of the architecture and pretraining strategy

In Figure 3, we plot the base model mean and worst group accuracy and DFR worst group accuracy for a wide range of models and pretraining strategies on each of the four image classification datasets. We provide model descriptions and training hyper-parameters in Appendix C. We train a total of 78 models on Waterbirds, 78 on CelebA, 40 on FMOW and 40 on CXR.

Accuracy on the line. Miller et al. [59] showed that for many distribution shifts in practice the out-of-distribution performance is highly correlated with the in-distribution generalization performance. In Figure 3 (top row), for each dataset we show scatter plots of base model mean accuracy vs base model worst group accuracy, analogously to Miller et al. [59]. For FMOW (which was also considered by Miller et al. [59]) we observe a linear correlation between the base model mean and worst group accuracies. On CXR, the dependence is also roughly linear. However, both on CelebA and on Waterbirds, the correlation does not appear entirely linear. In particular, on Waterbirds there is a large number of models that have similar worst group accuracy $\approx 20\%$, for which there appears to be little correlation between the base model WGA and mean accuracy. On CelebA, the same phenomenon occurs for the best performing models, with WGA between 40% and 50%.

DFR accuracy is on the line. Next, in the bottom panels of Figure 3, for each of the datasets we report the base model mean accuracy vs DFR WGA. On all datasets other than CXR[3], we observe a high linear correlation between the metrics, including the Waterbirds and CelebA. In particular, the models with the best mean (in-distribution) accuracy also achieve the best DFR WGA. Note that this is not the case for the base model WGA on CelebA, where the best base model WGA is 51% achieved by a ResNet-101 model pretrained on ImageNet1k; this model only achieves mean accuracy of 95.45% compared to 96.2% accuracy for the ConvNext XLarge model. The results in Figure 3 confirm that models that achieve better accuracy on the training data distribution generally learn better representations of the core features, and provide better worst group accuracies with DFR.

Are VITs more robust than CNNs? Ghosal et al. [20] noted that vision transformers pre-trained on ImageNet22k achieved better worst group accuracies on benchmarks with spurious correlations than popular CNN models. In particular, with a VIT-B/16 model, they achieve 89.3% worst group accuracy on Waterbirds. With the ConvNext Large model pretrained on ImageNet22k with ImageNet1k

[3] On CXR, we found that the base model is often hard to improve with DFR, suggesting that it already learns to weight the features correctly. We discuss the possible reasons and the nature of spurious features on this dataset in Appendix A.2.

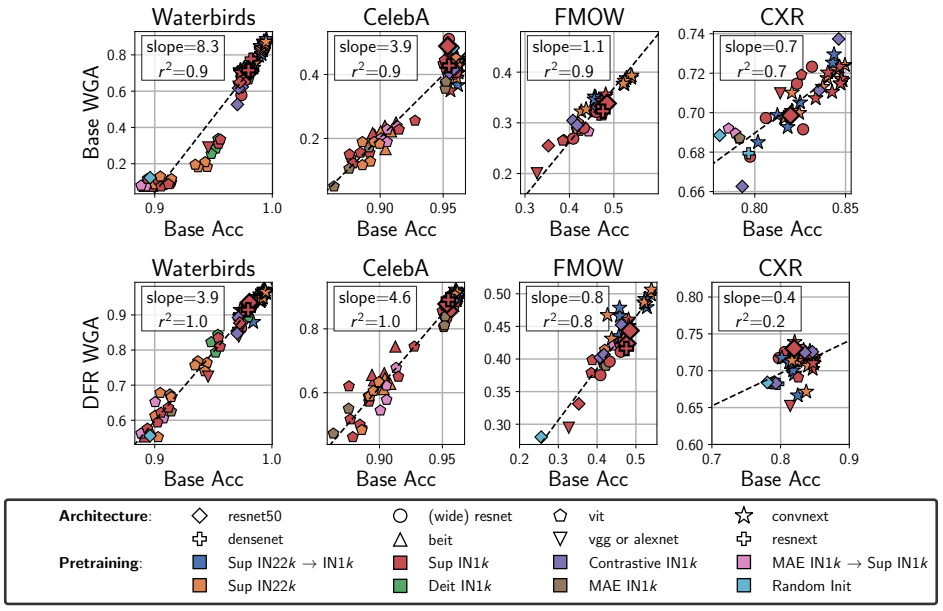


Figure 3: **Effect of the model architecture and pretraining.** Base model in-distribution accuracy plotted against base worst-group accuracy (top row) and DFR worst group accuracy (bottom row). For each panel, we additionally estimate the slope and the r^2 score for the linear fit to the data. On all datasets other than CXR, the DFR WGA is linearly correlated with base model in-distribution performance. For the base model WGA, the correlation with in-distribution performance is not entirely linear. Generally, models with better in-distribution performance provide better worst-group performance and better core feature representations.

finetuning, we achieve 88.9% worst group accuracy on the Waterbirds dataset. Notably, ConvNext is a CNN model and *not a vision transformer*. Generally, we observe that the models that provide the best in-distribution performance also provide better WGA. In our experiments, we did not observe qualitative differences between the results for vision transformers and CNN models.

ERM features are sufficient for SOTA performance. The DFR WGA results for the Waterbirds, CelebA and FMOW datasets significantly improve upon the previous best reported results, to the best of our knowledge. In particular, the ConvNext Large model pretrained on ImageNet22k with ImageNet1k finetuning achieves 97.2% DFR WGA on Waterbirds and 92.2% on CelebA; on FMOW, we only considered smaller ConvNext versions due to computational constraints, still achieving 50.6% DFR WGA with ConvNext Small pretrained on ImageNet22k; the current best results on the WILDS leaderboard for this dataset are 47.6% followed by 35.5% [43]. We note that our DFR evaluation uses the validation set to train the last layer of the model, similarly to e.g. Nam et al. [63], and unlike most standard group robustness methods which only use the validation set to tune the parameters. However, the results presented in this section prove that standard ERM with a strong pretrained model can achieve outstanding results on the robustness benchmarks, significantly improving upon specialized group robustness methods using a weaker model.

Strong in-distribution generalization correlates with improved robustness to spurious correlations (measured by the DFR worst group accuracy), and this trend holds regardless of the underlying architecture (CNN or ViT).

6.2 Does training on the target data improve features?

Above, we have shown that the choice of the base model architecture and pretraining has a large effect on the quality of the learned feature representations, as measured by DFR WGA. It is then natural to ask how much of the feature learning happens during training on the target data, and how much is simply transferred from the pretraining task. To answer this question, for the models trained

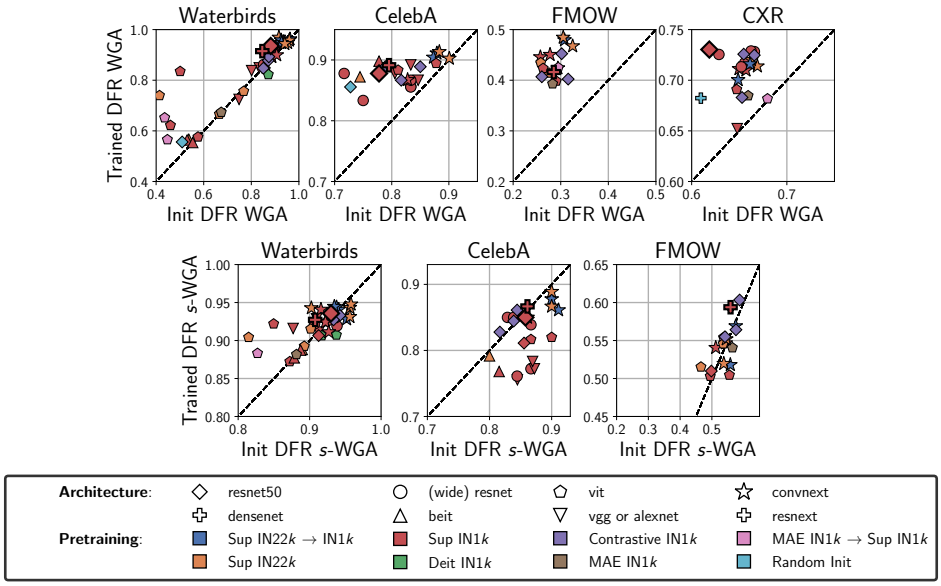


Figure 4: **Effect of training on target data.** DFR WGA (top 4 panels) and DFR s-WGA (bottom 3 panels) before and after training. On Waterbirds, the results do not significantly improve from training on the target data. On the other datasets, the performance improves consistently after training on the target data. Interestingly, on CelebA the DFR s-WGA decreases during training for many models, meaning that the spurious gender feature becomes less predictable from the learned representations.

in the previous section, we run the DFR evaluation on the initial weights of the models, without training the feature extractor on the target data. We report the DFR WGA and DFR s-WGA results in Figure 4. We repeat this experiment on the MultiNLI dataset in Appendix Table 5.

Surprisingly, we find that on Waterbirds, for most models the improvement from training on the target (Waterbirds) data is very small, if any. For example, for the ImageNet1k-pretrained ResNet-50 model that was not trained on Waterbirds data at all[4], we get 88.2% worst group accuracy by simply training the last layer on the validation data with DFR. If we finetune the feature extractor on the Waterbirds training data, we can achieve 92.9% DFR WGA. For reference, the state-of-the-art group DRO method achieves 91% WGA on Waterbirds with this architecture.

Furthermore, with the ConvNext Large model pretrained on ImageNet22k, we get 94% DFR WGA without training the feature extractor on the Waterbirds data, exceeding the best results previously reported in the literature, to the best of our knowledge. This strong performance is not particularly surprising, as ImageNet22k has many of the Waterbirds bird types as classes. The performance is almost unchanged by training on the target data: DFR WGA after training is 94.3%. From these results, we can conclude that Waterbirds performance should not be used as a primary metric for feature learning performance, especially if large-scale pretraining is used! Indeed, it is possible to achieve outstanding performance, exceeding the previously reported state-of-the-art, without training the features on the target data.

On the other datasets, the feature learning is more pronounced: the DFR WGA improves after training for all the models considered. However, on CelebA it is still possible to achieve 88.3% WGA without training on CelebA data, with ConvNext XLarge pretrained on ImageNet22k, while the best result that we were able to achieve with feature extractors trained on CelebA is 92.2%.

Is the spurious feature representation improved during training? We additionally explore the quality of representation of the spurious feature via DFR s-WGA. We show the results in Figure 4 (bottom row). Interestingly, on CelebA the spurious gender feature becomes less predictable during training for many of the models. On FMOW and Waterbirds there is no consistent trend, and the spurious feature does not become significantly more or less predictable during training.

[4]This model is used as initialization by most works that report experiments on the Waterbirds dataset.

Strong feature extractors trained on large-scale datasets are sufficient for outstanding performance on benchmark spurious correlation datasets, without any finetuning on the target data. Finetuning can improve the quality of the learned representations but often only to a relatively small degree (e.g. Waterbirds).

6.3 Effect of pretraining strategy

Finally, using a fixed ResNet-50 model architecture we evaluate the effect of pretraining strategy. On each of the four datasets, we train a ResNet-50 model initialized with (1) random initialization, (2) supervised pretraining, (3) DINO pretraining [10], (4) SimCLR pretraining [11] and (5) Barlow Twins pretraining [96] on ImageNet1k. We report the results in Figure 5. On Waterbirds and FMOW datasets, the randomly initialized model does not provide competitive performance. On CelebA, it still underperforms the pretrained models, but the gap is much smaller. Among all pretraining methods, supervised is preferable, but the contrastive methods are highly competitive.

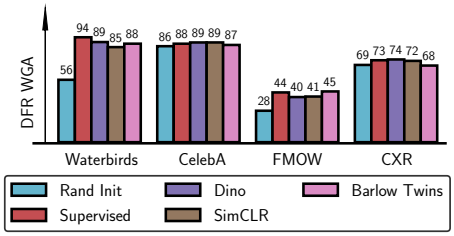


Figure 5: **Effect of pretraining.** Supervised pretraining provides the best performance, but contrastive methods are competitive. Random initialization performs poorly.

In Appendix Table 4 on MultiNLI with BERT models, we also show that pretraining is crucial for strong performance, while the specific choice of pretraining data has a smaller effect. In Appendix D we provide further ablations exploring the effect of weight decay and data augmentation on the learned feature representations. We find that while appropriate regularization can be helpful, models trained with no weight decay and no data augmentation can achieve highly competitive DFR WGA, especially on the standard Waterbirds and CelebA benchmarks.

ImageNet pretraining improves the representation of core features on many image datasets with spurious correlations.

7 Discussion

The worst group performance of a model is affected by two factors: the quality of the representation of the core features produced by the feature extractor and the weight assigned to the core features in the last classification layer. In contrast to prior work, we consider the quality of the feature extractor in isolation, focusing on realistic datasets and large-scale models. We find that many of the popular group robustness methods improve the worst group performance primarily by learning a better last layer and not by learning a better feature representation. Similarly, regularization techniques such as early stopping and strong weight decay can improve the worst group accuracy by learning a better last layer, but do not lead to a consistent improvement in terms of the quality of the learned feature representations. On the other hand, the base model architecture and pre-training strategy have a major effect on the quality of the feature representations.

Our observations suggest an important open question: is it possible to significantly improve upon standard ERM in terms of the quality of the learned representations for a given base model? In the future work, we will evaluate methods such as Rich Feature Construction [99], gradient starvation [68], ensembling [47] and other techniques for increasing feature diversity [e.g. 48]. We hope that our work will also inspire new group robustness methods targeted specifically at improving the quality of the core feature representations.

Acknowledgements. This research is supported by NSF CAREER IIS-2145492, NSF I-DISRE 193471, NIH R01DA048764-01A1, NSF IIS-1910266, NSF 1922658 NRT-HDR: FUTURE Foundations, Translation, and Responsibility for Data Science, NSF Award 1922658, Meta Core Data Science, Google AI Research, BigHat Biosciences, Capital One, and an Amazon Research Award. This work was supported in part through the NYU IT High Performance Computing resources, services, and staff expertise.

References

[1] Agarwal, A., Beygelzimer, A., Dudík, M., Langford, J., and Wallach, H. (2018). A reductions approach to fair classification. In *International Conference on Machine Learning*, pages 60–69. PMLR. (Cited on page 28)

[2] Alcorn, M. A., Li, Q., Gong, Z., Wang, C., Mai, L., Ku, W.-S., and Nguyen, A. (2019). Strike (with) a pose: Neural networks are easily fooled by strange poses of familiar objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4845–4854. (Cited on page 2)

[3] Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. (2019). Invariant risk minimization. *arXiv preprint arXiv:1907.02893*. (Cited on page 6)

[4] Bahng, H., Chun, S., Yun, S., Choo, J., and Oh, S. J. (2020). Learning de-biased representations with biased representations. In *International Conference on Machine Learning*, pages 528–539. PMLR. (Cited on page 6)

[5] Balestriero, R., Bottou, L., and LeCun, Y. (2022). The effects of regularization and data augmentation are class dependent. *arXiv e-prints*, pages arXiv–2204. (Cited on page 26)

[6] Bao, H., Dong, L., and Wei, F. (2021). Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*. (Cited on page 22)

[7] Biewald, L. (2020). Experiment tracking with weights and biases. Software available from wandb.com. (Cited on page 18)

[8] Borkan, D., Dixon, L., Sorensen, J., Thain, N., and Vasserman, L. (2019). 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. (Cited on page 4, 19)

[9] Brendel, W. and Bethge, M. (2019). Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. *arXiv preprint arXiv:1904.00760*. (Cited on page 2)

[10] Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. (2021). Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660. (Cited on page 10, 22)

[11] Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR. (Cited on page 10, 22)

[12] Christie, G., Fendley, N., Wilson, J., and Mukherjee, R. (2018). Functional map of the world. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. (Cited on page 4, 18)

[13] Creager, E., Jacobsen, J.-H., and Zemel, R. (2021). Environment inference for invariant learning. In *International Conference on Machine Learning*, pages 2189–2200. PMLR. (Cited on page 2, 5)

[14] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*. (Cited on page 5)

[15] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*. (Cited on page 7, 22)

[16] Dwork, C., Hardt, M., Pitassi, T., Reingold, O., and Zemel, R. (2012). Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226. (Cited on page 28)

[17] Eisenstein, J. (2022). Informativeness and invariance: Two perspectives on spurious correlations in natural language. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4326–4331. (Cited on page 3)

[18] Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., and Wichmann, F. A. (2020). Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673. (Cited on page 1)

[19] Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., and Brendel, W. (2018). Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*. (Cited on page 1, 2)

[20] Ghosal, S. S., Ming, Y., and Li, Y. (2022). Are vision transformers robust to spurious correlations? *arXiv e-prints*, pages arXiv–2203. (Cited on page 2, 7)

[21] Goyal, P., Duval, Q., Reizenstein, J., Leavitt, M., Xu, M., Lefaudeux, B., Singh, M., Reis, V., Caron, M., Bojanowski, P., Joulin, A., and Misra, I. (2021). Vissl. https://github.com/facebookresearch/vissl. (Cited on page 18)

[22] Gururangan, S., Swayamdipta, S., Levy, O., Schwartz, R., Bowman, S. R., and Smith, N. A. (2018). Annotation artifacts in natural language inference data. *arXiv preprint arXiv:1803.02324*. (Cited on page 1, 2)

[23] Hardt, M., Price, E., and Srebro, N. (2016). Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29. (Cited on page 28)

[24] Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825):357–362. (Cited on page 18)

[25] He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. (2021a). Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*. (Cited on page 22)

[26] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778. (Cited on page 5, 22)

[27] He, P., Gao, J., and Chen, W. (2021b). Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*. (Cited on page 27)

[28] He, P., Liu, X., Gao, J., and Chen, W. (2020). Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*. (Cited on page 27)

[29] Hendrycks, D., Mu, N., Cubuk, E. D., Zoph, B., Gilmer, J., and Lakshminarayanan, B. (2019). Augmix: A simple data processing method to improve robustness and uncertainty. *arXiv preprint arXiv:1912.02781*. (Cited on page 22, 26)

[30] Hermann, K., Chen, T., and Kornblith, S. (2020). The origins and prevalence of texture bias in convolutional neural networks. *Advances in Neural Information Processing Systems*, 33:19000–19015. (Cited on page 2)

[31] Hermann, K. and Lampinen, A. (2020). What shapes feature representations? exploring datasets, architectures, and training. *Advances in Neural Information Processing Systems*, 33:9995–10006. (Cited on page 2)

[32] Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708. (Cited on page 5, 22)

[33] Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95. (Cited on page 18)

[34] Idrissi, B. Y., Arjovsky, M., Pezeshki, M., and Lopez-Paz, D. (2021). Simple data balancing achieves competitive worst-group-accuracy. *arXiv preprint arXiv:2110.14503*. (Cited on page 2, 4, 5, 19, 24)

[35] Kang, B., Xie, S., Rohrbach, M., Yan, Z., Gordo, A., Feng, J., and Kalantidis, Y. (2019). Decoupling representation and classifier for long-tailed recognition. *arXiv preprint arXiv:1910.09217*. (Cited on page 3)

[36] Kaushik, D., Hovy, E., and Lipton, Z. (2020). Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations*. (Cited on page 3)

[37] Kaushik, D. and Lipton, Z. C. (2018). How much reading does reading comprehension require? a critical investigation of popular benchmarks. *arXiv preprint arXiv:1808.04926*. (Cited on page 2)

[38] Kaushik, D., Setlur, A., Hovy, E., and Lipton, Z. C. (2021). Explaining the efficacy of counterfactually augmented data. *International Conference on Learning Representations (ICLR)*. (Cited on page 3)

[39] Khani, F., Raghunathan, A., and Liang, P. (2019). Maximum weighted loss discrepancy. *arXiv preprint arXiv:1906.03518*. (Cited on page 28)

[40] Kirichenko, P., Izmailov, P., and Wilson, A. G. (2022). Last layer re-training is sufficient for robustness to spurious correlations. *arXiv preprint arXiv:2204.02937*. (Cited on page 1, 2, 3, 4, 5, 6, 7, 24, 25)

[41] Kleinberg, J., Mullainathan, S., and Raghavan, M. (2016). Inherent trade-offs in the fair determination of risk scores. *arXiv preprint arXiv:1609.05807*. (Cited on page 28)

[42] Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., and Willing, C. (2016). Jupyter notebooks – a publishing format for reproducible computational workflows. In Loizides, F. and Schmidt, B., editors, *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, pages 87 – 90. IOS Press. (Cited on page 18)

[43] Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Gao, I., et al. (2021). Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR. (Cited on page 2, 4, 8, 18, 19)

[44] Kolesnikov, A., Beyer, L., Zhai, X., Puigcerver, J., Yung, J., Gelly, S., and Houlsby, N. (2020). Big transfer (bit): General visual representation learning. In *European conference on computer vision*, pages 491–507. Springer. (Cited on page 2)

[45] Kolesnikov, A. and Lampert, C. H. (2016). Improving weakly-supervised object localization by micro-annotation. *arXiv preprint arXiv:1605.05538*. (Cited on page 2)

[46] Krizhevsky, A. (2014). One weird trick for parallelizing convolutional neural networks. *arXiv preprint arXiv:1404.5997*. (Cited on page 22)

[47] Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30. (Cited on page 10)

[48] Lee, Y., Yao, H., and Finn, C. (2022). Diversify and disambiguate: Learning from underspecified data. *arXiv preprint arXiv:2202.03418*. (Cited on page 2, 4, 5, 6, 7, 10, 19, 28)

[49] Li, Y., Li, Y., and Vasconcelos, N. (2018). Resound: Towards action recognition without representation bias. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 513–528. (Cited on page 2)

[50] Liu, E. Z., Haghgoo, B., Chen, A. S., Raghunathan, A., Koh, P. W., Sagawa, S., Liang, P., and Finn, C. (2021). Just train twice: Improving group robustness without training group information. In *International Conference on Machine Learning*, pages 6781–6792. PMLR. (Cited on page 2, 4, 5, 7)

[51] Liu, Z., Luo, P., Wang, X., and Tang, X. (2015). Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738. (Cited on page 4, 18)

[52] Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., and Xie, S. (2022). A convnet for the 2020s. *arXiv preprint arXiv:2201.03545*. (Cited on page 2, 22)

[53] Loshchilov, I. and Hutter, F. (2017). Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*. (Cited on page 24, 28)

[54] Lovering, C., Jha, R., Linzen, T., and Pavlick, E. (2020). Predicting inductive biases of pre-trained models. In *International Conference on Learning Representations*. (Cited on page 3)

[55] Marcel, S. and Rodriguez, Y. (2010). Torchvision the machine-vision package of torch. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 1485–1488. (Cited on page 18)

[56] McCoy, R. T., Pavlick, E., and Linzen, T. (2019). Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. *arXiv preprint arXiv:1902.01007*. (Cited on page 1, 2)

[57] McKinney, W. (2010). Data structures for statistical computing in python. In van der Walt, S. and Millman, J., editors, *Proceedings of the 9th Python in Science Conference*, pages 51 – 56. (Cited on page 18)

[58] Menon, A. K., Rawat, A. S., and Kumar, S. (2020). Overparameterisation and worst-case generalisation: friend or foe? In *International Conference on Learning Representations*. (Cited on page 3)

[59] Miller, J. P., Taori, R., Raghunathan, A., Sagawa, S., Koh, P. W., Shankar, V., Liang, P., Carmon, Y., and Schmidt, L. (2021). Accuracy on the line: on the strong correlation between out-of-distribution and in-distribution generalization. In *International Conference on Machine Learning*, pages 7721–7735. PMLR. (Cited on page 7)

[60] Mo, S., Kang, H., Sohn, K., Li, C.-L., and Shin, J. (2021). Object-aware contrastive learning for debiased scene representation. *Advances in Neural Information Processing Systems*, 34. (Cited on page 2)

[61] Moayeri, M., Pope, P., Balaji, Y., and Feizi, S. (2022). A comprehensive study of image classification model sensitivity to foregrounds, backgrounds, and visual attributes. *arXiv preprint arXiv:2201.10766*. (Cited on page 2)

[62] Nam, J., Cha, H., Ahn, S., Lee, J., and Shin, J. (2020). Learning from failure: De-biasing classifier from biased classifier. *Advances in Neural Information Processing Systems*, 33:20673–20684. (Cited on page 2)

[63] Nam, J., Kim, J., Lee, J., and Shin, J. (2022). Spread spurious attribute: Improving worst-group accuracy with spurious attribute estimation. In *International Conference on Learning Representations*. (Cited on page 2, 4, 7, 8)

[64] Niven, T. and Kao, H.-Y. (2019). Probing neural network comprehension of natural language arguments. *arXiv preprint arXiv:1907.07355*. (Cited on page 2)

[65] Oakden-Rayner, L., Dunnmon, J., Carneiro, G., and Ré, C. (2020). 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. (Cited on page 2, 4, 5, 18, 19)

[66] Pagliardini, M., Jaggi, M., Fleuret, F., and Karimireddy, S. P. (2022). Agree to disagree: Diversity through disagreement for better transferability. *arXiv preprint arXiv:2202.04414*. (Cited on page 2)

[67] Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in pytorch. (Cited on page 18)

[68] Pezeshki, M., Kaba, O., Bengio, Y., Courville, A. C., Precup, D., and Lajoie, G. (2021). Gradient starvation: A learning proclivity in neural networks. *Advances in Neural Information Processing Systems*, 34. (Cited on page 10)

[69] Pleiss, G., Raghavan, M., Wu, F., Kleinberg, J., and Weinberger, K. Q. (2017). On fairness and calibration. *Advances in neural information processing systems*, 30. (Cited on page 28)

[70] Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., Ding, D., Bagul, A., Langlotz, C., Shpanskaya, K., et al. (2017). Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. *arXiv preprint arXiv:1711.05225*. (Cited on page 4, 5, 19)

[71] Robbins, H. and Monro, S. (1951). A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407. (Cited on page 24)

[72] Rosenfeld, A., Zemel, R., and Tsotsos, J. K. (2018). The elephant in the room. *arXiv preprint arXiv:1808.03305*. (Cited on page 2)

[73] Rosenfeld, E., Ravikumar, P., and Risteski, A. (2022). Domain-adjusted regression or: Erm may already learn features sufficient for out-of-distribution generalization. *arXiv preprint arXiv:2202.06856*. (Cited on page 3)

[74] Ruan, Y., Dubois, Y., and Maddison, C. J. (2021). Optimal representations for covariate shift. *arXiv preprint arXiv:2201.00057*. (Cited on page 6)

[75] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252. (Cited on page 4, 5)

[76] Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. (2019). Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*. (Cited on page 2, 4, 5, 7, 18, 23, 24, 26, 27)

[77] Sagawa, S., Koh, P. W., Lee, T., Gao, I., Xie, S. M., Shen, K., Kumar, A., Hu, W., Yasunaga, M., Marklund, H., et al. (2021). Extending the wilds benchmark for unsupervised adaptation. *arXiv preprint arXiv:2112.05090*. (Cited on page 4, 18, 19)

[78] Sagawa, S., Raghunathan, A., Koh, P. W., and Liang, P. (2020). An investigation of why overparameterization exacerbates spurious correlations. In *International Conference on Machine Learning*, pages 8346–8356. PMLR. (Cited on page 2)

[79] Shah, H., Tamuly, K., Raghunathan, A., Jain, P., and Netrapalli, P. (2020). The pitfalls of simplicity bias in neural networks. *Advances in Neural Information Processing Systems*, 33:9573–9585. (Cited on page 1)

[80] Shetty, R., Schiele, B., and Fritz, M. (2019). Not using the car to see the sidewalk–quantifying and controlling the effects of context in classification and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8218–8226. (Cited on page 2)

[81] Shi, Y., Daunhawer, I., Vogt, J. E., Torr, P. H., and Sanyal, A. (2022). How robust are pre-trained models to distribution shift? *arXiv preprint arXiv:2206.08871*. (Cited on page 3)

[82] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*. (Cited on page 22)

[83] Singla, S. and Feizi, S. (2021). Salient imagenet: How to discover spurious features in deep learning? *arXiv preprint arXiv:2110.04301*. (Cited on page 2)

[84] Sohoni, N., Sanjabi, M., Ballas, N., Grover, A., Nie, S., Firooz, H., and Ré, C. (2021). Barack: Partially supervised group robustness with guarantees. *arXiv preprint arXiv:2201.00072*. (Cited on page 2, 7)

[85] Teney, D., Abbasnejad, E., Lucey, S., and Hengel, A. v. d. (2021). Evading the simplicity bias: Training a diverse set of models discovers solutions with superior ood generalization. *arXiv preprint arXiv:2105.05612*. (Cited on page 2, 28)

[86] Veitch, V., D'Amour, A., Yadlowsky, S., and Eisenstein, J. (2021). Counterfactual invariance to spurious correlations: Why and how to pass stress tests. *arXiv preprint arXiv:2106.00545*. (Cited on page 3)

[87] Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272. (Cited on page 18)

[88] Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. (2011). The caltech-ucsd birds-200-2011 dataset. (Cited on page 18)

[89] Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., and Summers, R. M. (2017). Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2097–2106. (Cited on page 4, 18)

[90] Wightman, R. (2019). Pytorch image models. `https://github.com/rwightman/pytorch-image-models`. (Cited on page 18)

[91] Williams, A., Nangia, N., and Bowman, S. R. (2017). A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*. (Cited on page 4, 29)

[92] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. (2019). Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*. (Cited on page 18)

[93] Xiao, K., Engstrom, L., Ilyas, A., and Madry, A. (2020). Noise or signal: The role of image backgrounds in object recognition. *arXiv preprint arXiv:2006.09994*. (Cited on page 1, 2)

[94] Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. (2017). Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500. (Cited on page 22)

[95] Zagoruyko, S. and Komodakis, N. (2016). Wide residual networks. *arXiv preprint arXiv:1605.07146*. (Cited on page 22)

[96] Zbontar, J., Jing, L., Misra, I., LeCun, Y., and Deny, S. (2021). Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR. (Cited on page 10, 22)

[97] Zech, J. R., Badgeley, M. A., Liu, M., Costa, A. B., Titano, J. J., and Oermann, E. K. (2018). Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: a cross-sectional study. *PLoS medicine*, 15(11):e1002683. (Cited on page 1, 2)

[98] Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. (2017). mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*. (Cited on page 22, 26)

[99] Zhang, J., Lopez-Paz, D., and Bottou, L. (2022). Rich feature construction for the optimization-generalization dilemma. *arXiv preprint arXiv:2203.15516*. (Cited on page 6, 10)

[100] Zhang, M., Sohoni, N. S., Zhang, H. R., Finn, C., and Ré, C. (2021). Correct-n-contrast: A contrastive approach for improving robustness to spurious correlations. In *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*. (Cited on page 2, 6, 7)

[101] Zhong, Z., Zheng, L., Kang, G., Li, S., and Yang, Y. (2020). Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13001–13008. (Cited on page 22, 26)

[102] Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., and Torralba, A. (2017). Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464. (Cited on page 18)

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] Limitations are described in the appendix.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] Broader impact is descussed in the appendix.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A] No theoretical results.
   (b) Did you include complete proofs of all theoretical results? [N/A] No theoretical results.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] We report errorbars for the main experimental results in the appendix.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We include an estimate of the total compute used in the appendix.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes] We mention the license of all assets in the appendix.
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A] Not including new assets.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] Not including new assets
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] Not including new assets
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] No research on human subjects or crowdsourcing.
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] No research on human subjects or crowdsourcing.
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] No research on human subjects or crowdsourcing.

Appendix Outline

This appendix is structured as follows. In Section A we describe the datasets, augmentation policies and models used in this paper. In Section B we provide details on the methods, implementations and hyper-parameters, as well as detailed results for the experiments in Section 5. In Section C we provide additional details on the experiments in Section 6. In Section D we provide results on the effect of regularization and data augmentation. In Section E we provide additional results on the MultiNLI dataset. Finally, in Section F we describe the limitation, broader impact, compute and licenses.

Tools and packages. During the work on this paper, we used the following tools and packages: `NumPy` [24], `SciPy` [87], `PyTorch` [67], `TorchVision` [55], `Jupyter notebooks` [42], `Matplotlib` [33], `Pandas` [57], `Weights&Biases` [7], `timm` [90], `transformers` [92], `vissl` [21].

A Data and Models

In this section, we describe the datasets, data augmentation policies and models used throughout the paper.

A.1 Datasets

We perform experiments on 4 image classification and 2 text classification problems. We illustrate the image datasets in Figure 6 and the text datasets in Figure 7.

Waterbirds. The Waterbirds dataset is described in Figure 6. The dataset contains images of birds from the CUB dataset [88] pasted on the backgrounds from the Places dataset [102]. The spurious attribute s describes the type of background (water or land) and the core feature associated with the target y is the type of the bird (waterbird or landbird). For a detailed description of the data generating process, see Sagawa et al. [76]. The background is spuriously correlated with the bird type in the training datasets: waterbirds are more likely to be placed on a water background, and landbirds are more likely to be placed on a land background. There are 4 groups defined to the tuples (y, s).

CelebA. The CelebA hair color dataset is described in Figure 6. The dataset contains photos of celebrities from the CelebA dataset Liu et al. [51]. The core attribute associated with the target y is the hair color (blond vs non-blond). The gender serves as a spurious feature s: the vast majority of blond people in CelebA are female. There are 4 groups defined to the tuples (y, s).

FMOW. The WILDS-FMOW dataset is described in Figure 6. This dataset is a part of the WILDS benchmark [43, 77], and was originally collected by Christie et al. [12]. The dataset contains satellite images, and the target y describes the type of building or land use shown in the image. There are 62 classes. The spurious attribute s corresponds to the region (Asia, Europe, Africa, America, Oceania) shown in the image. The training data additionally contains another group *Other*, which is dropped during evaluation. For this dataset, following the WILDS benchmark, we define groups by just the value of the spurious attribute: $g = s$. In particular, worst group accuracy corresponds to the worst accuracy across regions. The regions are represented unequally in the data, leading to unequal performance. Moreover, the test images were taken several years later than the train images, constituting an additional type of distribution shift. We note that for the DFR evaluation, we use the validation data to train the last layer of the model, which means that our results would be classified as non-standard submissions to the WILDS leaderboard.

CXR. The CXR dataset is described in Figure 6. The images are taken from the CXR-14 dataset [89]. CXR-14 is a multi-label dataset, where each label corresponds to a disease, and one image can show multiple diseases. We perform the pneumothorax classification task, i.e. all images that have the pneumothorax label have $y = 1$ and all images that do not have $y = 0$. The dataset contains several images for some of the patients; there is no patient overlap between the train, test and validation splits. Oakden-Rayner et al. [65] identified a hidden stratification in this dataset: a lot of images showing patients with pneumothorax showed a *chest drain*, which is a treatment for the pneumothorax disease. The neural networks trained on this dataset are using the chest drain as a shortcut feature, and perform much worse when classifying images without the chest drain. The labels for the spurious feature are only available on the validation and test datasets, and only for the images showing sick patients, i.e. there is no $(y = 0, s = 1)$ group. There are 3 groups corresponding to available pairs (y, s).

Following prior work [e.g. 65, 48, 70], we compute three AUC values: for classifying group 0 against group 1, group 0 against group 2 and group 0 against the combined groups 0 and 1; instead of worst group accuracy we report the lower of the first two AUC values, and instead of mean accuracy we report the last AUC value throughout the experiments.

Civil Comments The Civil Comments Coarse dataset is described in Figure 7. The dataset was originally collected in Borkan et al. [8] and is a part of the WILDS benchmark [43, 77]. This dataset contains comments that are classified as toxic or not toxic. We use the coarse version of the dataset, following Idrissi et al. [34]. The spurious attribute s determines whether or not the comment mentions one of the following protected attributes: male, female, LGBT, black, white, Christian, Muslim, other religions. These protected attributes are mentioned more frequently in toxic comments compared to neutral comments, constituting a spurious correlation. There are 4 groups corresponding to the pairs (y, s).

MultiNLI The MultiNLI dataset is described in Figure 7. It contains pairs of sentences, and the label y describes the relationship between the sentences: contradiction, entailment or neutral. The spurious attribute s describes the presence of negation words, which appear much more frequently in the examples from the negation class. There are 6 groups corresponding to the pairs (y, s).

A.2 The nature of the spurious correlations

The nature of the spurious correlation differs between the datasets that we consider. On Waterbirds, CelebA, Civil Comments and MultiNLI, the spurious attribute is correlated with the target on train, but is not generally predictive of the class label across the groups. In particular, we wish to train a model that ignores the spurious attribute in its predictions, as using the spurious attribute *hurts* the performance on some of the groups. By applying DFR on a group-balanced validation set, we try to train a model that ignores the spurious feature.

On the FMOW dataset, the groups correspond to the spurious attribute (region), and not pairs (y, s). As the regions are not represented equally, standard ERM is incentivised to perform well on the majority groups, with less weight on the minority groups. In this case, we do not wish to remove the reliance on the spurious attribute, but instead we wish to find a model that performs well on the minority groups. We apply DFR to this dataset by retraining the last layer on a group-balanced validation set, analogously to the other datasets.

Finally, in the CXR dataset, there are no examples showing healthy patients with a chest drain (to the best of our knowledge). Consequently, the reliance on the spurious attribute s does not necessarily *hurt* performance on the other groups, unlike e.g. on the Waterbirds dataset. In this case, we do not wish to remove ignore the feature s, but instead we wish to find a model that performs well on the images with $s = 0$ (no chest drain). Applying DFR to this dataset is not straightforward, as (1) we do not have examples from the $(y = 0, s = 1)$ group, and (2) we do not wish to remove the reliance on the s attribute. We found that the best approach in this case was to simply train a logistic regression model on all of the available validation data, without any balancing. DFR generally provides a smaller improvement on this dataset, and behaves less consistently, as we see e.g. in Figure 3.

Waterbirds			**Target**: bird type;	**Spurious feature**: background type.
Image:				
Group g**:**	0	1	2	3
Target y**:**	0	0	1	1
Spurious s**:**	0	1	0	1
Description:	landbird on land	landbird on water	waterbird on land	waterbird on water
# Train data:	3498 (73%)	184 (4%)	56 (1%)	1057 (22%)
# Val data:	467	466	133	133

CelebA hair color			**Target**: hair color;	**Spurious feature**: gender.
Image:				
Group g**:**	0	1	2	3
Target y**:**	0	0	1	1
Spurious s**:**	0	1	0	1
Description:	non-blond woman	non-blond man	blond woman	blond man
# Train data:	71629 (44%)	66874 (41%)	22880 (14%)	1387 (1%)
# Val data:	8535	8276	2874	182

Wilds-FMOW			**Target**: land use / building;	**Spurious feature**: region.	
Image:					
Group g**:**	0	1	2	3	4
Target y**:**	$\{0,\ldots,61\}$	$\{0,\ldots,61\}$	$\{0,\ldots,61\}$	$\{0,\ldots,61\}$	$\{0,\ldots,61\}$
Spurious s**:**	0	1	2	3	4
Description:	Asia	Europe	Africa	America	Oceania
# Train data:	17809 (23%)	34816 (45%)	1582 (2%)	20973 (27%)	1641 (2%)
# Val data:	4121	7732	803	6562	693

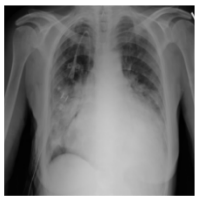
CXR-14			**Target**: pneumothorax;	**Shortcut feature**: chest drain.
Image:				
Group g**:**	0	1	2	
Target y**:**	0	1	1	
Spurious s**:**	0	0	1	
Description:	not sick, no chest drain	sick, no chest drain	sick, chest drain	
# Train data:	71629 (95%)	?	?	
# Val data:	10714	204	300	

Figure 6: **Image datasets.** Group descriptions and example images for Waterbirds, CelebA hair color, Wilds-FMOW and CXR-14 datasets. Each column corresponds to a group in the dataset. For CXR-14 group labels are not known on the training data.

Civil Comments				**Target**: toxic / neutral comment; **Spurious feature**: mentions protected categories.	
Example	**Group** g	**Target** y	**Spur.** s	**# Train data**	**# Val data**
"I wouldn't think this would be so rare on the plains of eastern Colorado."	0	0 (Netral)	0	148186 (55%)	25159
"If the person wanted to write to the Bishop, he or she would have. They wanted the Vatican to know their pain, not to acknowledge it is a lack of Christian charity and kindness."	1	0 (Netral)	1	90337 (33%)	14966
"What a gross example of bureaucrats and lawyers showing everyone in Oregon who our bosses are. Next the jerks will demand we bow to them. <...>"	2	1 (Toxic)	0	12731 (5%)	2111
"Democrats, RINO's and atheists won't be happy until they have destroyed conservatives and christians."	3	1 (Toxic)	1	17784 (7%)	2944

MultiNLI				**Target**: contradiction / entailment / neutral; **Spurious feature**: has negation words.	
Example	**Group** g	**Target** y	**Spur.** s	**# Train data**	**# Val data**
"he was up quickly. [SEP] he sat the entire time and didn't move."	0	0 (contr.)	0	57498 (28%)	22814
"for golf enthusiasts, two courses reassure the visitor you are close to civilization. [SEP] there is no golf course available anywhere around."	1	0 (contr)	1	11158 (5%)	4634
"while emergency physicians may not have the time or interest, the patients do. [SEP] the patients have time and interest, unlike emergency physicians."	2	1 (entail.)	0	67376 (32%)	26949
"then it dawned on him that of course the lawyer did not know. [SEP] he realised that the lawyer had no idea."	3	1 (entail.)	1	1521 (1%)	613
"disneyland is huge and can be very crowded in summer. [SEP] going to disneyland is every child's dream.	4	2 (neutr.)	0	66630 (32%)	26655
"you have raced him, senor? " he asked drew with formal courtesy. [SEP] drew replied that he had never raced him."	5	2 (neutr.)	1	1992 (1%)	797

Figure 7: **Text datasets.** Text examples, class labels, spurious attributes and group labels for the Civil Comments and MultiNLI datasets. On both datasets, the spurious feature is correlated with the class label.

A.3 Data augmentation and preprocessing

On the image datasets, we consider several data augmentation policies. In Figure 8, we provide the code implementing the **No augmentation** and **Default augmentation** policies in torchvision. The other policies are more involved, so we do not include the code here. All policies normalize the data analogously to the code in Figure 8.

No augmentation. This policy resizes the data to a fixed resolution and applies channel normalization.

Default policy. This policy additionally applies random crops and horizontal flips.

MixUp. For this policy, we use the Default policy to initially preprocess the images, and then apply MixUp [98] with the mixing parameter $\alpha = 0.2$.

Random Erasing. This policy is described in Zhong et al. [101]. It randomly erases rectangular blocks of the image, and replaces them with uniform grey blocks. We use the implementation in `timm.data.random_erasing` in the `timm` package.

Augmix. We adapt the official implementation of the AugMix policy [29] available here.

Text models. For data preprocessing in the experiments on text data, we use the BERT tokenizer: `BertTokenizer.from_pretrained("bert-base-uncased")` from the `transformers` package.

A.4 Models

Here, we list the models used in this paper.

ResNet-50. On the Waterbirds, CelebA and FMOW by default we use the ResNet-50 model He et al. [26] pretrained on ImageNet. We use the model implemented in the `torchvision` package: `torchvision.models.resnet50(pretrained=True)`. In Figure 5, we additionally consider the randomly initialized model `torchvision.models.resnet50(pretrained=False)`, and the models pre-trained with SimCLR [11] and Barlow Twins [96] contrastive learning methods, imported from the `vissl` package (models available here).

DenseNet-121. On CXR, by default we use the DenseNet-121 model [32] implemented in the `torchvision` package: `torchvision.models.densenet121(pretrained=True)`. We also consider this model without ImageNet pretraining: `torchvision.models.densenet121(pretrained=False)`.

Other image models. In Section 6, we additionally consider a broad range of architectures and pretraining methods, which we briefly list here. We use the following models from `torchvision`, all pretrained on ImageNet1k: ResNet-18, ResNet-34, ResNet-50, ResNet-101, ResNet-152 [26]; Wide-ResNet-50-2 [95]; ResNext-50-32 $\times$ $4d$ [94]; DenseNet-121 [32]; VGG-16, VGG-19 [82]; AlexNet [46]. We also use the following models from the `timm` package: ConvNext-Small, ConvNext-Base, ConvNext-Large, ConvNext-XLarge [52], pretrained on either ImageNet1k or ImageNet22k; ViT-Small, ViT-Base, ViT-Large, ViT-Huge [15], pretrained on either ImageNet1k or ImageNet22k; BEiT-Base, BEiT-Large [6], pretrained on either ImageNet1k or ImageNet22k; DEiT-Small, DEiT-Base [6], pretrained ImageNet1k. We also use ViT-Small, ViT-Base models with DINO pretraining on ImageNet1k [10] available here. Finally, we use ViT-Base, ViT-Large and ViT-Huge models with MAE pretraining on ImageNet1k [25], with or without supervised finetuning on ImageNet1k, available here.

BERT model. On the text classification problems, we use the BERT for classification model from the `transformers` package: `BertForSequenceClassification.from_pretrained('bert-base-uncased', num_labels=num_classes)`.

B Details: ERM vs Group Robustness

B.1 Methods and hyper-parameters

ERM, RWY and RWG. We report the hyper-parameters used on each of the datasets in Table 2. We did not tune the hyper-parameters for ERM, RWY and RWG aside from the learning rate for the text classification problems. For all vision datasets, we used the default data augmentation policy (see Section A.3). We describe the default model choices in Section A.4. The RWY method is

```python
import torchvision.transforms as transforms
import torch


target_resolution = (224, 224)
resize_resolution = (256, 256)
IMAGENET_STATS = ([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])


# No Augmentation
noaug_transform = transforms.Compose(
    transforms.RandomResizedCrop(
        target_resolution,
        scale=(0.7, 1.0),
        ratio=(0.75, 1.33),
        interpolation=2),
    transforms.RandomHorizontalFlip(),
    transforms.ToTensor(),
    transforms.Normalize(*IMAGENET_STATS)
    )

# Default Augmentation
aug_transform = transforms.Compose(
    transforms.Resize(resize_resolution),
    transforms.CenterCrop(target_resolution)
    transforms.ToTensor(),
    transforms.Normalize(*IMAGENET_STATS)
    )

# On test, we always use noaug_transform
test_transform = noaug_transform
```

Figure 8: **Data augmentation policies.** The default data augmentation policies implemented using the `torchvision` package.

implemented by providing a `sampler` to the `DataLoader` in `PyTorch`, which samples the datapoints from different classes with the same frequency. The RWG method is implemented analogously, and samples datapoints from different groups with the same frequency.

Group DRO. The training objective used in Sagawa et al. [76] is the following:

$$\hat{\theta} = \arg\min_{\theta} \max_{g \in \mathcal{G}} \left[\mathbb{E}_{(x,y)\sim p_g} l(y, f_\theta(x)) + C/\sqrt{n_g} \right],$$

where $f_\theta(\cdot)$ is a neural network model with parameters θ, $l(\cdot,\cdot)$ is a loss function (cross-entropy for classification), $\mathcal{G}$ is a set of all groups, n_g is the size of the group g, and C is a generalization adjustment hyper-parameter. We follow Sagawa et al. [76] to choose hyper-parameter combinations for tuning group DRO on Waterbirds, CelebA and MultiNLI.

In particular, on Waterbirds we considered the following combinations of the initial learning rate lr and weight decay wd: $(lr = 10^{-3}, wd = 10^{-4})$, $(lr = 10^{-4}, wd = 0.1)$ and $(lr = 10^{-5}, wd = 1)$. We varied the parameter C in the range $\{0, 1, 2, 3, 4, 5\}$ for each combination of learning rate and weight decay. Additionally, we varied weight decay in range $\{0, 10^{-4}, 10^{-4}, 3 \cdot 10^{-4}, 10^{-3}, 3 \cdot 10^{-3}, 10^{-2}, 3 \cdot 10^{-2}, 0.1, 0.3, 1\}$ for fixed values of learning rate 10^{-3} and $C = 0$.

On CelebA, we considered the following combinations of lr and wd: $(lr = 10^{-4}, wd = 10^{-4})$, $(lr = 10^{-4}, wd = 10^{-2})$ and $(lr = 10^{-5}, wd = 0.1)$, varying C in the same range as on Waterbirds. For weight decay ablation we considered values in $\{0, 10^{-5}, 3 \cdot 10^{-5}, 10^{-4}, 3 \cdot 10^{-4}, 10^{-3}, 3 \cdot 10^{-3}, 10^{-2}\}$.

On MultiNLI, we use learning rate $2 \cdot 10^{-5}$, and vary weight decay in the range $\{0., 0.01, 0.1, 1.0\}$ and C in the range $\{0, 1, 3, 5\}$.

Method	Waterbirds	CelebA	FMOW	CXR	MultiNLI	CivilComments
ERM	$68.9_{\pm 2.0}$	$44.0_{\pm 2.1}$	$31.4_{\pm 0.7}$	$68.8_{\pm 0.2}$	$67.5_{\pm 1.1}$	$61.0_{\pm 0.3}$
ERM + DFR	$91.1_{\pm 0.8}$	$89.4_{\pm 0.9}$	$41.6_{\pm 0.6}$	$71.6_{\pm 0.5}$	$72.6_{\pm 0.3}$	$78.8_{\pm 0.5}$
RWY	$65.4_{\pm 0.6}$	$46.1_{\pm 2.1}$	$30.5_{\pm 0.6}$	$71.3_{\pm 1.1}$	$68.0_{\pm 0.4}$	$63.4_{\pm 0.9}$
RWY + DFR	$90.4_{\pm 1.0}$	$88.3_{\pm 0.5}$	$40.9_{\pm 0.7}$	$72.5_{\pm 0.1}$	$72.2_{\pm 1.9}$	$78.5_{\pm 0.4}$
RWY-ES	$74.5_{\pm 0.0}$	$76.8_{\pm 7.7}$				$78.9_{\pm 1.0}$
RWY-ES + DFR	$89.1_{\pm 0.7}$	$89.6_{\pm 0.5}$				$76.9_{\pm 0.6}$
RWG	$67.7_{\pm 0.7}$	$49.1_{\pm 0.9}$	$31.2_{\pm 0.1}$		$62.0_{\pm 0.2}$	$73.5_{\pm 2.2}$
RWG + DFR	$91.3_{\pm 0.3}$	$85.4_{\pm 1.5}$	$41.1_{\pm 0.6}$		$71.6_{\pm 1.3}$	$79.3_{\pm 0.5}$
RWG-ES	$77.4_{\pm 0.0}$	$83.7_{\pm 0.7}$				$74.7_{\pm 6.7}$
RWG-ES + DFR	$90.7_{\pm 0.2}$	$89.8_{\pm 0.3}$				$77.0_{\pm 0.2}$
GDRO	$68.5_{\pm 6.0}$	$66.3_{\pm 7.8}$	30.2		70.1	70.6
GDRO + DFR	$88.2_{\pm 1.1}$	$90.4_{\pm 0.7}$	40.3		71.8	80.2
GDRO-ES	$90.7_{\pm 0.6}$	$90.6_{\pm 1.6}$	33.1		73.5	80.4
GDRO-ES + DFR	$89.9_{\pm 0.5}$	$91.1_{\pm 0.1}$	42.5		73.3	77.3

Table 1: **Method comparison results.** Detailed results for the method comparison presented in Figure 1. The error bars represent one standard deviation over 3 independent runs. We only evaluate early stopping (ES) on datasets where it is helpful for the base model performance. On CXR, no group information is available on the train data, so we can only apply the ERM and RWY methods. We use a ResNet50 model pretrained on ImageNet on Waterbirds, CelebA and FMOW, a DenseNet-121 model pretrained on ImageNet on CXR, and a BERT model pretrained on Book Corpus and English Wikipedia data on CivilComments and MultiNLI.

Dataset	Optimizer	Initial LR	LR schedule	Batch size	Weight decay	# Epochs
Waterbirds	SGD [71]	$3 \cdot 10^{-3}$	Cosine annealing	32	10^{-4}	100
CelebA	SGD [71]	$3 \cdot 10^{-3}$	Cosine annealing	100	10^{-4}	20
FMOW	SGD [71]	$3 \cdot 10^{-3}$	Cosine annealing	100	10^{-4}	20
CXR	SGD [71]	$3 \cdot 10^{-3}$	Cosine annealing	100	10^{-4}	20
MultiNLI	AdamW [53]	10^{-5}	Linear annealing	16	10^{-4}	10
Civil Comments	AdamW [53]	10^{-5}	Linear annealing	16	10^{-4}	10

Table 2: **ERM, RWY and RWG hyper-parameters.** Default hyper-parameters used on each dataset. On the image classification datasets, we adapted the hyper-parameters of Kirichenko et al. [40], with no tuning. On the text classification dataset, we followed Sagawa et al. [76] and Idrissi et al. [34] in the choice of the optimizer and learning rate scheduler, and chose the learning rate value which provided the best base model validation mean accuracy.

On CivilComments, we considered the following combinations of lr and wd: $(lr = 10^{-5}, wd = 0.1)$, $(lr = 10^{-5}, wd = 0.01)$, $(lr = 10^{-4}, wd = 0.01)$; for each combination we varied C in $\{0, 3, 5\}$.

On FMOW, the learning rate and weight decay pairs were $(lr = 10^{-3}, wd = 10^{-3})$, $(lr = 10^{-4}, wd = 10^{-3})$, $(lr = 10^{-4}, wd = 10^{-2})$, $(lr = 10^{-5}, wd = 10^{-1})$. For each lr and wd combination we varied C in the same range as on Waterbirds and CelebA datasets.

For all image datasets, we use the default data augmentation policy (see Section A.3). In all runs we used the same batch size and train for the same number of epochs as the corresponding hyper-parameters in ERM (see Table 2).

B.2 DFR implementation and hyper-parameters

In all experiments, we use the $\text{DFR}^{\text{Val}}_{\text{Tr}}$ variation of the DFR method described in Kirichenko et al. [40]. We follow the official implementation provided here. Specifically, we use ℓ_1 regu-

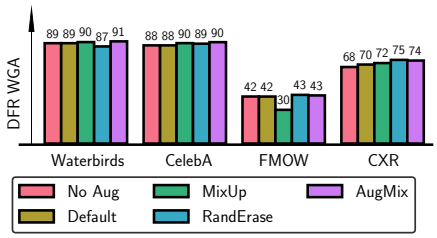


Figure 9: **Effect of data augmentation.** All the results shown use a ResNet50 architecture except for CXR results, which use a DenseNet-121 model pretrained on ImageNet. AugMix augmentation policy consistently provides strong performance across all datasets, while random erasing and mixup hurt performance on Waterbirds and FMOW respectively.

larization for training the logistic regression model implemented in the `scikit-learn` package: `sklearn.LogisticRegression(penalty="l1", C=c, solver="liblinear")`; we tune the regularization strength c within the set $\{1, 0.7, 0.3, 0.1, 0.07, 0.03, 0.01\}$, following the procedure described in Kirichenko et al. [40].

As explained in Section A.2, on CXR we train the logistic regression model on all of the validation set without group balancing. Further, on CXR we tune the regularization strength parameter according to the worst AUC, and not worst group accuracy.

We additionally compute DFR s-WGA by using DFR (with the same hyper-parameters and implementation) to predict the spurious attribute s (instead of the class label y) from the learned features.

B.3 Results

We provide detailed results for all methods on all datasets in Table 1. Group-DRO significantly improves the base model performance compared to all other methods across the board. After applying DFR, the performance across the different methods is very similar, although Group-DRO with early stopping still typically provides a small improvement. The improvement is, however, very small compared to the improvement from using a better base model (see Section 6).

C Details: Effect of the Base Model

For all experiments in Section 6, we use the default data augmentation policy (see Section A.3). We consider a broad range of models and architectures (see Section A.4). We ran all models with the default hyper-parameters provided in Table 2. For CXR-14 dataset, we used class reweighting due to heavy class imbalance present in the train data (95% of train images are from the negative class).

We note that the default hyper-parameters are suboptimal for some of the ViT models, which lead to poor performance for some of the models. With more tuning, we expect that it should be possible to improve the results further for all the considered models, especially ViT-based.

D Effect of Regularization

Regularization is used to combat overfitting, including reliance on spurious features. We consider two regularization techniques: weight decay and data augmentation.

Effect of weight decay. Using a ResNet-50 model pretrained on ImageNet1k, we run training with a range of weight decay values on each of the four image datasets, and report the results in Figure 10. We use the default model setup described in Section A.4, and default hyper-parameters in Table 2, and vary the weight decay strength in the range $\{0, 10^{-5}, 10^{-4}, 3 \cdot 10^{-4}, 10^{-3}, 3 \cdot 10^{-3}, 10^{-2}\}$. For the base model WGA performance, it is generally helpful to set the weight decay to non-zero values (in particular, on Waterbirds, CelebA and FMOW datasets). Indeed, on CelebA the base model WGA for no weight decay is the worst across all weight decay values, losing to the best weight decay by $\approx 5\%$. However, the no weight decay model is in fact the best model according to DFR WGA

on CelebA! On Waterbirds and CelebA, strong weight decay allows the model to rely less on the spurious features in the last layer (leading to higher base WGA), but does not improve the learned feature representations (similar or worse DFR WGA). This observation is especially interesting given that Sagawa et al. [76] showed that group DRO requires stronger than usual weight decay to achieve good performance.

At the same time, on FMOW and CXR non-zero weight decay appears to be helpful for learning high quality representations of the core features. We hypothesize that the difference in weight decay effects on these two pairs of datasets is due to the difference in the nature of distribution shifts. In particular, Waterbirds and CelebA are standard benchmarks for spurious correlations robustness and group robustness, where for each data point both core and spurious feature are present and clear. CXR and FMOW are more challenging datasets with real-world distribution shift. The groups in FMOW are defined by the regions which are not equally represented in train data, but the spurious features (region identity) are more subtle and not present in all images. Moreover, FMOW additionally contains a domain shift across time: the validation and test images were taken after 2016, while the training images were taken before 2016. On CXR the spurious feature is only present for the positive class and completely absent for the negative class (see Section A.2 for detailed discussion). Consequently, we observe some differences in the results between the standard benchmarks and these more realistic datasets.

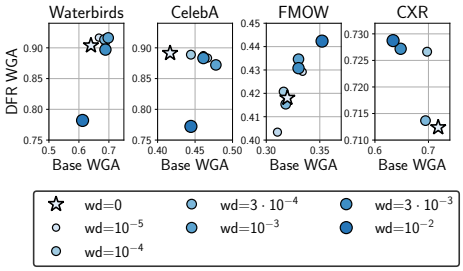


Figure 10: **Effect of weight decay.** While zero weight decay underperforms in base WGA on Waterbirds and CelebA, it provides near-optimal DFR WGA.

In Table 6 we report analogous results on the MultiNLI text classification problem.

Effect of Data Augmentation. Next, we consider 5 data augmentation policies: (1) no augmentations, (2) default augmentations, i.e. random crops and horizontal flips, (3) MixUp [98] combined with default augmentations, (4) Random Erasing [101], and (5) AugMix [29]. We train a ResNet-50 model pretrained on ImageNet1k on each of the four datasets with each of the augmentation policies, and report the results in Figure 9. We use the default models described in Appendix A.4, and default hyper-parameters in Table 2, and apply the data augmentation policies desribed in Appendix A.3.

AugMix provides the best performance on each dataset. However, we find that data augmentation is generally not required to achieve strong performance on any of the datasets with the exception of CXR: the model trained without augmentations is competitive across the board. Moreover, data augmentation can *hurt* the learned features. For example, while MixUp is helpful on Waterbirds and CelebA, it significantly hurts the preformance on FMOW, with 30% DFR WGA compared to 42% for the model trained without augmentation. We hypothesize that on the FMOW dataset, which contains highly detailed satellite images, mixing the images makes training overly challenging.

Similarly, Random Erasing hurts the DFR WGA on Waterbirds. We hypothesize that the randomly erased image block is more likely to fully cover the bird features than the background features, as the bird occupies a small fraction of the image relative to the background. Consequently, the model is incentivised to focus on the spurious feature: the model trained with Random Erasing learned the highest quality representation of the spurious feature across all augmentations with DFR s-WGA of $91.8 \pm 0.1\%$ across 3 runs, compared to e.g. $91.4 \pm 0.5\%$ for the model trained with no augmentation.

Balestriero et al. [5] report a related observation for models trained on ImageNet: in some cases data augmentation affects different classes disproportionately, and best data augmentation policies for mean accuracy can lead to poor worst-class accuracy,

Effect of training length on ERM. We plot the DFR WGA, DFR s-WGA, as well as base model WGA and mean accuracy as a function of training epoch for ERM training in Figure 11. On all datasets, we observe that the DFR WGA quickly converges and stays roughly constant throughout training. On all datasets, 5 epochs is sufficient for near-optimal performance. Longer training generally does not help or hurt DFR WGA, even when it hurts the base model.

Effect of training length on Group-DRO. We repeat the same experiment, but with Group-DRO instead of ERM training in Figure 12. Again, we observe that 5 epochs are generally sufficient for

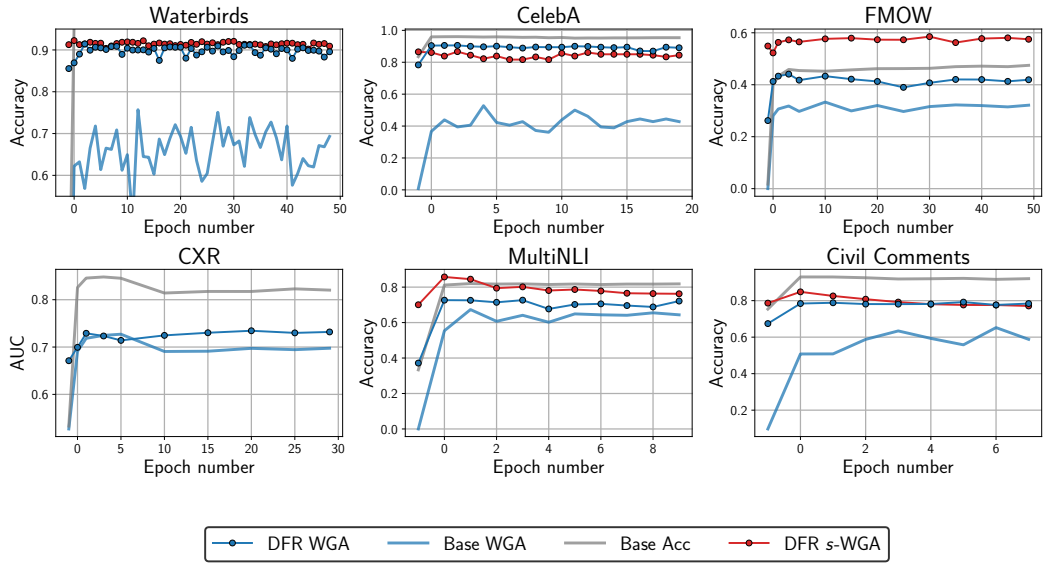


Figure 11: **ERM training length effect.** On all datasets, 5 epochs or less is sufficient to achieve near-optimal DFR WGA performance, but longer training does not hurt performance.

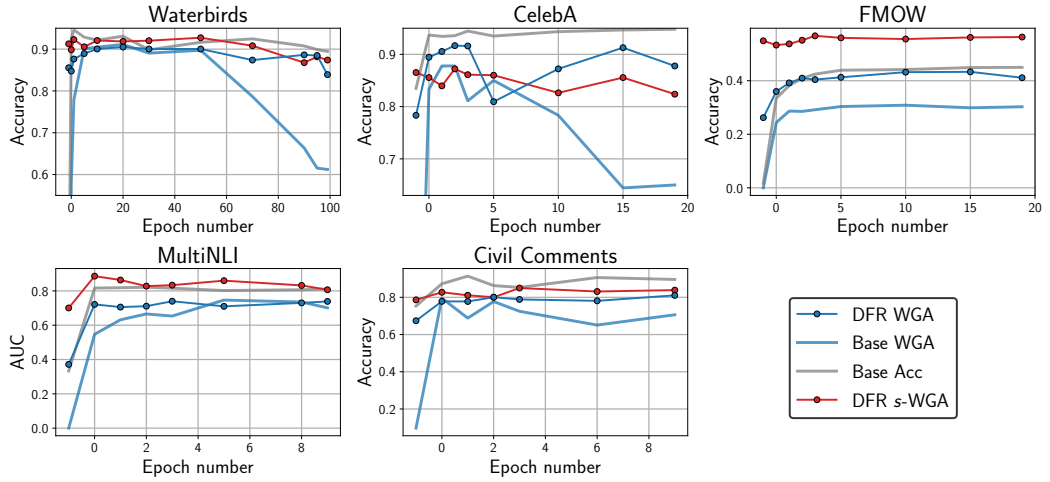


Figure 12: **Group-DRO training length effect.** DFR WGA converges to near-optimal performance in under 5 epochs. On Waterbirds and CelebA the DFR WGA performance is less stable compared to the ERM results in Figure 11.

near-optimal performance. Interestingly, for Group-DRO, DFR WGA does deteriorate over time on Waterbirds but not as significantly as the base model WGA.

E Additional results on MultiNLI

For all experiments in this section, we train the models for 5 epochs with learning rate 10^{-5} and 0 weight decay.

Effect of base model. In Table 3, we report the results for BERT-Base, BERT-Large, DeBERTa-Base, and DeBERTa-Large models [27, 28]. For BERT models, following Sagawa et al. [76], we use the cached tokenizer outputs with maximum sequence length 150. For DeBERTa models, we re-tokenize the dataset with the corresponding tokenizer with a maximum sequence length of

Base Model	Base Acc	Base WGA	DFR WGA	DFR s-WGA
BERT-Base	82.3	66.1	73.8	83.6
BERT-Large	84.6	70.6	76.6	79.0
DeBERTa-Base	88.9	80.4	82.3	80.2
DeBERTa-Large	90.2	81.2	84.8	68.3

Table 3: **MultiNLI: base model effect.** Effect of base model on the performance on the MultiNLI dataset. DeBERTa-Large model provides the best performance in terms of DFR WGA as well as Base WGA and Base mean accuracy.

Pretraining	Base Acc	Base WGA	DFR WGA	DFR s-WGA
Random Init	59.1	26.0	45.4	94.9
Pretrained on Wiki + Book Corpus	82.3	66.1	73.8	83.6
MultiLingual on Wiki	80.5	62.1	72	83.9

Table 4: **MultiNLI: effect of pretraining.** Results for BERT-Base model with different types of pretraining on the MultiNLI dataset. Pretraining is required to achieve strong performance. Multilingual pretraining is competitive but inferior to pretraining on data in English.

220. We train all models for 5 epochs. The advanced DeBERTa-Large model provides the best base performance and DFR WGA.

Effect of pretraining. In Table 4 we evaluate the results of BERT-Base models with different types of pretraining on MultiNLI. This experiment is analogous to the experiment for image classification problems presented in Figure 5(a). We find that pretraining is necessary to achieve strong performance on this dataset, but different pretraining datasets lead to competitive results.

Effect of training on target data. In Table 5 we evaluate the effect of training on the MultiNLI dataset for the BERT and DeBERTa pretrained models. This experiment is analogous to the experiment for image classification problems presented in Figure 4. We find that after training on the target data both the core features (DFR WGA) and the spurious features (DFR s-WGA) become significantly more decodable. This result is in contrast to the results on Waterbirds in Figure 4, where DFR WGA is not significantly improved from training.

Effect of weight decay. In Table 6 we report the results of the weight decay ablation for the BERT-Base model on MultiNLI; this model uses the AdamW optimizer [53], so we consider larger values of weight decay.

F Broader Impact and Limitations

Limitations. While we consider a wide range of factors that affect the feature learning under spurious correlations, we inevitably do not cover all the possible factors. In particular, it would be interesting to consider the effect of regularization methods beyond weight decay and early stopping, and methods for diverse feature learning such as DivDis [48], or the method of Teney et al. [85]. As another limitation, while DFR performs well in our experiments, it is not guaranteed to learn an optimal linear classifier with the given features; further improvements in learning the last layer can be used to refine the results of our study. Despite these limitations, we believe that our work provides a comprehensive analysis of the feature learning under spurious correlations.

Broader impact. Research on spurious correlations is closely related to ML Fairness [16, 23, 41, 69, 1, 39]. We hope that our work can motivate further research in fairness, where techniques similar to DFR can be considered to improve the fairness of ML models. A potential negative outcome that can result from *misinterpretation* of our analysis is if the practitioners assume that spurious correlations are not an important issue, as ERM learns high quality representation of the core features. We emphasize that ERM still performs suboptimally (see Figure 1), as it does not provide a correct weighting for the features in the final classification layer. Spurious correlations are a significant practical issue that should be considered carefully in real-world applications.

Pretraining	Init DFR WGA	Trained DFR WGA	Init DFR s-WGA	Trained DFR s-WGA
BERT-Base	37.4	73.8	69.9	83.6
BERT-Large	34.4	76.6	57.6	81
DeBERTa-Base	58.5	82.3	77	80
DeBERTa-Large	61.8	84.8	66.4	68.3

Table 5: **MultiNLI: effect of training on target data.** Results for different models before and after training on the MultiNLI dataset. For all the considered models, both the core and the spurious features are significantly more decodabe after training on the target data.

Weight decay	Base Acc	Base WGA	DFR WGA	DFR s-WGA
0	82.3	66.1	73.8	83.6
1	81.4	63.2	74.6	90.1
3	74.5	43.0	66.0	94.4
10	63.5	23.4	39.9	66.0
30	57.0	15.2	28.9	59.3
100	46.4	1.7	3.2	53.0

Table 6: **MultiNLI: weight decay.** The effect of weight decay on the BERT-Base model on MultiNLI. Similarly to the results in Figure 10, weight decay 0 provides competitive performance. The best DFR WGA is achieved with weight decay 1.

Compute. We estimate the total compute used in the process of working on this paper at roughly 3000 GPU hours. The compute usage is dominated by the experiments presented in Figure 3, where we trained a large number of large-scale models on 4 vision datasets. The tuning of Group-DRO hyper-parameters was also relatively compute-heavy. The experiments were run on GPU clusters on Nvidia Tesla V100, Titan RTX, RTX8000, 3080 and 1080Ti GPUs.

Licenses. The Civil Comments dataset is distributed under the CC0 license. The FMOW dataset is under the FMoW Challenge Public License. The Places dataset is under the CC BY license. For the details of the license for the MultiNLI dataset, see Williams et al. [91].