DISTRIBUTIONALLY ROBUST NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Overparameterized neural networks trained to minimize average loss can be highly accurate *on average* on an i.i.d. test set, yet consistently fail on atypical groups of the data (e.g., by learning spurious correlations that do not hold at test time). Distributionally robust optimization (DRO) provides an approach for learning models that instead minimize *worst-case* training loss over a set of pre-defined groups. We find, however, that naively applying DRO to overparameterized neural networks fails: these models can perfectly fit the training data, and any model with vanishing average training loss will also already have vanishing worst-case training loss. Instead, the poor worst-case performance of these models arises from poor *generalization* on some groups. As a solution, we show that increased regularization—e.g., stronger-than-typical weight decay or early stopping—allows DRO models to achieve substantially higher worst-group accuracies, with $10\% - 40\%$ improvements over standard models on a natural language inference task and two image tasks, while maintaining high average accuracies. Our results suggest that regularization is critical for worst-group performance in the overparameterized regime, even if it is not needed for average performance. Finally, we introduce and provide convergence guarantees for a stochastic optimizer for this group DRO setting, underpinning the empirical study above.

1 INTRODUCTION

Machine learning models are typically trained to maximize average accuracy on a test set that is independent and identically distributed (i.i.d.) to the training set. However, even models that perform well on average can consistently fail on rare and atypical examples (Hovy & Sgaard, 2015; Blodgett et al., 2016; Tatman, 2017; Hashimoto et al., 2018; Duchi et al., 2019). Such models are problematic if, e.g., they violate equity considerations (Jurgens et al., 2017; Buolamwini & Gebru, 2018) or rely on *spurious correlations*, which are misleading heuristics that work for most training examples but might not hold up on a different test distribution. For example, in natural language inference (NLI)—determining if two sentences agree or contradict—negation words like 'never' tend to be strongly correlated with contradiction due to artifacts in crowdsourced annotation of training data (Gururangan et al., 2018). A model could achieve high average accuracy on an i.i.d. test set by learning the correlation between negation and contradictions, but would perform poorly on test sets where that spurious correlation does not hold (McCoy et al., 2019).

We show in this paper that in the modern regime of training overparameterized models—i.e., large neural networks that can perfectly fit the training data—this problem is one of generalization: while these models can generalize well *on average* (as in Zhang et al. (2017)), they do not generalize well on the *worst-case* (sub)group (e.g., the group of contradictions with no negations).

To avoid the pitfalls of optimizing for average loss, we instead optimize for the worst-case loss over various groups in the data, using prior knowledge of spurious associations (e.g., between negation and contradiction) to choose these groups. This is an instance of distributionally robust optimization (DRO), which studies worst-case performance over potential test distributions (Ben-Tal et al., 2013; Duchi et al., 2016). In our *group DRO* setting, the data is distributed as a mixture of different groups with mixture weights that can vary at test time (Hu et al., 2018; Oren et al., 2019). Optimizing for the worst-case test distribution is thus equivalent to minimizing the worst-case loss over each group.

Prior work has applied DRO to learn models that are robust over different groups of data (Duchi & Namkoong, 2018; Hashimoto et al., 2018) or as a data-dependent regularizer (Maurer & Pontil,

Figure 1: Representative training and test examples for the datasets we consider. The correlation between the label y and the spurious attribute a at training time does not hold at test time.

2009; Shafieezadeh-Abadeh et al., 2015; Duchi & Namkoong, 2016). However, a major challenge with this approach in the modern overparameterized regime is that models can achieve vanishing training loss (Zhang et al., 2017)); any such model would be (near) optimal for both the average and worst-case training losses, so there is little incentive for their minimizers to differ (Wen et al., 2014). Indeed, existing work on DRO has focused on either generative models or convex predictive models with limited capacities, both of which cannot get vanishing training loss.

In this paper, we study group DRO in the context of large, high-performing neural networks in three applications (Figure 1)—natural language inference with the MultiNLI dataset (Williams et al., 2018), facial attribute recognition with CelebA (Liu et al., 2015), and bird photograph recognition with our modified version of the CUB dataset (Wah et al., 2011). In the vanishing-training-loss regime, we find that group DRO models do no better than ERM models: both models achieve nearly perfect training accuracies across groups and continue to perform well on average at test time, but suffer low test accuracies on the worst-case group (Section 3.1). In other words, the generalization gap is small on average, but the worst-case generalization gap over groups is large.

However, group DRO models can significantly outperform ERM models when we account for these generalization gaps—through appropriate regularization, e.g., strong weight decay or early stopping (Section 3.2), and through *group adjustments* that explicitly handle the differences in generalization gaps between groups (Section 3.3). Across the three applications, group DRO improves worst-case test accuracies by 10% to 40% while maintaining comparably high average test accuracies. These results give a new perspective on generalization in neural networks: regularization might not be important for good average performance (models can, e.g., "train longer and generalize better" on average (Hoffer et al., 2017)) but it appears critical for good worst-case performance.

Finally, we introduce a stochastic optimizer for group DRO that underpins the experiments above and scales to large models and datasets. We derive convergence guarantees and rates for our algorithm in the convex case, and empirically show it behaves well in the non-convex case (Section 5).

2 Setup

Consider the setting where we wish to predict labels $y \in \mathcal{Y}$ from input features $x \in \mathcal{X}$. Given a model family Θ, loss $\ell : \Theta \times (\mathcal{X} \times \mathcal{Y}) \to \mathbb{R}_+$, and training data drawn from some distribution P, the standard goal is to find a model $\theta \in \Theta$ that minimizes the expected loss under the same distribution $\mathbb{E}_P[\ell(\theta; (x, y)]$. This is typically done through empirical risk minimization (ERM):

$$\hat{\theta}_{\text{ERM}} := \arg\min_{\theta \in \Theta} \mathbb{E}_{(x,y)\sim\hat{P}}[\ell(\theta; (x, y))], \tag{1}$$

where $\hat{P}$ is the empirical distribution over the training data.

In distributionally robust optimization (DRO) (Ben-Tal et al., 2013; Duchi et al., 2016), we aim instead to minimize the worst-case loss over an uncertainty set of distributions $\mathcal{Q}$,

$$\min_{\theta \in \Theta} \Big\{ \mathcal{R}(\theta) := \sup_{Q \in \mathcal{Q}} \mathbb{E}_{(x,y)\sim Q}[\ell(\theta; (x, y))] \Big\}. \tag{2}$$

The uncertainty set $\mathcal{Q}$ encodes the possible test distributions that we want our model to perform well on, and it is typically chosen as a large divergence ball around the training distribution P. Choosing a general family $\mathcal{Q}$ confers robustness to a wide set of distributional shifts but can be too pessimistic because it optimizes for implausible worst-case scenarios (Hu et al., 2018; Oren et al., 2019).

Instead, following the above prior work on group DRO, we assume that the data-generating distribution P is a mixture of m different groups, $\mathcal{G} = \{1, 2, \dots, m\}$, with each $g \in \mathcal{G}$ defining a distribution P_g over (x, y). (In our experiments, $m = 4$ or 6.) We further assume that the test and training distributions can have arbitrarily different mixture weights; in other words, we choose $\mathcal{Q} = \{\sum_{g=1}^{m} q_g P_g : q \in \Delta_m\}$, where Δ_m is the $(m-1)$-dimensional probability simplex. Because the optimum of a linear program is always attained at a vertex, this choice of $\mathcal{Q}$ allows us to express the worst-case risk over $\mathcal{Q}$ as a maximum over the expected loss of each group

$$\mathcal{R}(\theta) = \max_{g \in \mathcal{G}} \mathbb{E}_{(x,y)\sim P_g}[\ell(\theta; (x, y))]. \tag{3}$$

To optimize $\mathcal{R}(\theta)$, we assume we additionally observe group identities at training—i.e., our training data comprises (x, y, g) triplets—but not at test time, so the model cannot use g directly. Instead, we use the training data to learn the *group DRO* model that minimizes the empirical robust risk $\hat{\mathcal{R}}(\theta)$:

$$\hat{\theta}_{\text{DRO}} := \arg\min_{\theta \in \Theta} \Big\{ \hat{\mathcal{R}}(\theta) := \max_{g \in \mathcal{G}} \mathbb{E}_{(x,y)\sim \hat{P}_g}[\ell(\theta; (x, y))] \Big\}, \tag{4}$$

where $\hat{P}_g$ is the empirical distribution over all training points (x, y, g') with $g' = g$. Group DRO learns models with good robust *training* loss across groups. This need not imply good robust *test* loss because of the *generalization gap* $\delta := \mathcal{R}(\hat{\theta}_{\text{DRO}}) - \hat{\mathcal{R}}(\hat{\theta}_{\text{DRO}})$. We will show that for overparameterized neural networks, δ is large and requires the use of regularizers.

2.1 Applications

In the rest of this paper, we study three applications that share a similar structure (Figure 1): each data point (x, y) has some input attribute $a(x) \in \mathcal{A}$ that is spuriously correlated with the label y, and we use this prior knowledge to form $m = |\mathcal{A}| \times |\mathcal{Y}|$ groups, one for each value of (a, y). We expect that models that learn the correlation between a and y in the training data would do poorly on groups for which $a \neq y$, and hence do worse on the robust loss $\mathcal{R}(\theta)$.

Object recognition with correlated backgrounds (Waterbirds dataset). Object recognition models can learn to make predictions from image backgrounds instead of the actual object (Ribeiro et al., 2016). We investigate this by constructing a new dataset, Waterbirds, which combines bird photographs from the Caltech-UCSD Birds-200-2011 (CUB) dataset (Wah et al., 2011) with image backgrounds from the Places dataset (Zhou et al., 2017). We label each bird as one of $\mathcal{Y} = \{\text{waterbird}, \text{landbird}\}$ and place it on one of $\mathcal{A} = \{\text{water background}, \text{land background}\}$, with waterbirds (landbirds) more frequently appearing against a water (land) background (Appendix B.1). There are $n = 4795$ training examples and 89 in the smallest group (waterbirds against land).

Object recognition with correlated demographics (CelebA dataset). Object recognition models (and other ML models more generally) have also been shown to learn spurious associations between the target label and demographic information like gender and ethnicity (Buolamwini & Gebru, 2018). We examine this on the CelebA celebrity face dataset (Liu et al., 2015), using hair color ($\mathcal{Y} = \{\text{blond}, \text{dark}\}$) as the target and gender ($\mathcal{A} = \{\text{male}, \text{female}\}$) as the spurious attribute. There are $n = 162770$ training examples, with 1387 in the smallest group (blond-haired males).

Natural language inference (MultiNLI dataset). In natural language inference, the task is to determine if a given hypothesis is entailed by, neutral with, or contradicts a given premise. Prior work has shown that crowdsourced training datasets for this task have significant annotation artifacts, such as the spurious correlation between contradictions and the presence of the negation words *nobody*, *no*, *never*, and *nothing* (Gururangan et al., 2018). We divide the MultiNLI dataset (Williams et al., 2018) into $m = 6$ groups, one for each pair of labels $\mathcal{Y} = \{\text{entailed}, \text{neutral}, \text{contradictory}\}$ and spurious attributes $\mathcal{A} = \{\text{no negation}, \text{negation}\}$. There are $n = 206175$ examples in the training set, with 1521 examples in the smallest group (entailment with negations); we modify the standard MultiNLI split to better estimate accuracy on small groups (Appendix B.1).

3 COMPARISON BETWEEN GROUP DRO AND ERM

To study the behavior of group DRO vs. ERM in a modern setting, we fine-tuned ResNet50 models (He et al., 2016) on Waterbirds and CelebA, and a BERT model (Devlin et al., 2018) on MultiNLI; these are popular and high-performing models for image and natural language tasks, respectively.

We train the ERM models using standard (minibatch) stochastic gradient descent. For the DRO models, we introduce a stochastic minibatch algorithm based on online mirror descent that optimizes for an adaptively-weighted training distribution at each iteration; we defer discussion of this to Section 5. We tune the learning rate for ERM and use the same setting for DRO (Appendix B.2).

For each model, we measure its *average* (in-distribution) accuracy over training and test sets drawn from the same distribution, as well as its *robust* accuracy over the worst-performing group.

3.1 ERM AND DRO HAVE POOR ROBUST ACCURACY IN THE OVERPARAMETERIZED REGIME

We start by examining the robust accuracy of models when they are trained to convergence using standard hyperparameter settings (He et al., 2016; Devlin et al., 2018).[1] These overparameterized models can attain near-perfect training accuracy and vanishing training loss even in the presence of default regularization (batch normalization and weight decay for ResNet50, and dropout for BERT).

ERM. As expected, ERM models reach near-perfect training accuracies of at least 99.8% on all three datasets, even on the worst-case group, and also obtain high average test accuracies (82.5%, 97.0%, and 94.8% on MultiNLI, Waterbirds, and CelebA). However, they perform poorly on the worst-case group at test time (66.4%, 55.4%, and 41.1%; Table 1, Figure 2). Their low robust accuracies imply that these models are brittle under distributional shifts; they are accurate only when the test and training distributions match.

DRO. In this vanishing-training-loss regime, there exist models that do almost perfectly on both the ERM (1) and DRO (4) objectives. We train group DRO models in the same way as above, and indeed find that they perform similarly to ERM models, attaining near-perfect training accuracies and high average test accuracies, but poor robust test accuracies (Table 1, Figure 2).

Discussion. The high average test accuracies of our ERM and DRO models are consistent with the widely-reported observation that neural networks can generalize well on average despite perfectly fitting the training data (Zhang et al., 2017). However, we observe that these models do *not* generalize well on the worst-case group, and consequently suffer from low robust accuracies. In other words, the gap between average and robust test accuracies arises not from poor robust training performance—the models are near-perfect at training time, even on the worst-case groups—but from variations in the generalization gaps across groups.

3.2 GROUP DRO IMPROVES ROBUST ACCURACY UNDER APPROPRIATE REGULARIZATION

Classically, regularization techniques control the generalization gap by constraining the model family's capacity to fit the training data. In the modern overparameterized regime, however, regularized models like those trained above can still perfectly fit the training data, and models still tend to do well even when all regularization is removed (Zhang et al., 2017).

Here, we explore increasing regularization strength until the models no longer perfectly fit the training data. We find that while average test accuracy remains high (as in the standard models trained above), the departure from the vanishing-training-loss regime allows group DRO models to significantly outperform ERM models on robust accuracy. We investigate two types of regularization:

Weight decay. The default weight decay strength (i.e., the coefficient of the L_2-norm penalty $\lambda\|\theta\|_2^2$) in ResNet50 is $\lambda = 0.0001$ (He et al., 2016). Increasing this by several orders of

[1] Training to convergence is a widespread practice for image models (Zhang et al., 2017; Hoffer et al., 2017). Pre-trained language models are typically pretrained until convergence (Devlin et al., 2018; Radford et al., 2019) but fine-tuned for a fixed small number of epochs because average test accuracy levels off quickly; we verified that training to convergence gave equally high average test accuracy.

			Average Accuracy		Robust Accuracy	
			ERM	DRO	ERM	DRO
Standard Regularization	Waterbirds	Train	100.0	100.0	100.0	100.0
		Test	97.0	97.0	55.4	65.6
	CelebA	Train	100.0	100.0	99.9	100.0
		Test	94.8	94.7	41.1	41.1
	MultiNLI	Train	99.9	99.4	99.8	99.0
		Test	82.5	82.0	66.4	66.4
Strong Weight Decay	Waterbirds	Train	97.0	98.9	37.1	97.1
		Test	95.8	96.1	21.5	86.1
	CelebA	Train	95.6	94.8	40.4	93.4
		Test	95.8	93.5	37.8	86.7
Early Stopping	Waterbirds	Train	86.0	82.0	9.0	73.8
		Test	93.3	92.7	5.5	84.9
	CelebA	Train	93.9	92.3	14.2	85.1
		Test	94.6	91.8	25.0	88.3
	MultiNLI	Train	92.1	86.1	78.6	83.3
		Test	82.8	81.4	66.0	77.7

Table 1: Average and robust accuracies for each training method. Both ERM and DRO models perform poorly on the worst-case group in the absence of regularization (top). With regularization (middle, bottom), DRO achieves high worst-group performance, significantly improving from ERM. Cells are colored by accuracy, from low (red) to medium (white) to high (blue) accuracy.

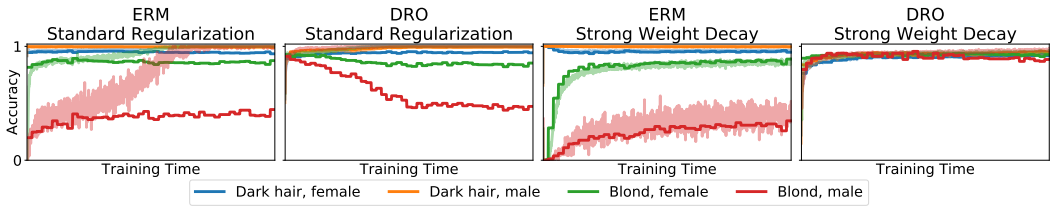

Figure 2: Training (light) and validation (dark) accuracy for CelebA throughout training. With default hyperparameters and training to convergence, ERM and DRO models achieve perfect training accuracy across groups, but generalize badly on the worst-case group (blond male, panels 1-2). With stronger weight decay, ERM models achieve high average train and test accuracies at the cost of the rare group (panel 3). DRO models achieve high train and test accuracies across groups (panel 4).

magnitude—to $\lambda = 1.0$ for Waterbirds and $\lambda = 0.1$ for CelebA—does two things: 1) it prevents both ERM and DRO models from achieving perfect training accuracy, and 2) substantially reduces the generalization gap for each group.

Both ERM and DRO models still achieve high average test accuracies that are comparable to each other and to the standard models with weaker regularization. However, because no model can achieve perfect training accuracy in this regime, ERM models sacrifice training accuracy on the worst-case group (robust accuracies of 37.1% and 40.4% for Waterbirds and CelebA; Table 1,Figure 2), and consequently obtain poor robust test accuracy.

In contrast, DRO models optimize for (and can still attain) high robust training accuracy (93.4% and 97.1% on Waterbirds and CelebA). The small generalization gap in this weight decay regime means that high robust training accuracy translates to high robust test accuracy: with group DRO, robust test accuracy improves from 21.5% to 86.1% on Waterbirds and from 37.8% to 86.7% on CelebA.

While these results show that strong weight decay has a striking impact on ResNet50 models for Waterbirds and CelebA, our initial experiments did not show a similar effect on BERT model for MultiNLI. Weight decay is not typically tuned for this task (Devlin et al., 2018)—our default hyperparameters set weight decay to zero (Appendix B.2)—so we turn to other forms of regularization.

Early stopping. A different, implicit form of regularization is early stopping (Hardt et al., 2015). We use the same settings in Section 3.1, but only train each model for a fixed (small) number of epochs (Section B.2). As with strong weight decay, curtailing training reduces the generalization gap and prevents models from fitting the data perfectly. In this setting, DRO also does substantially better than ERM on robust test accuracy, improving from 66.0% to 77.7% on MultiNLI, 5.5% to 84.9% on Waterbirds, and 25.0% to 88.3% on CelebA. Average test accuracies are comparably high in both ERM and DRO models, though there is a small drop of 1-2$\%$ for DRO (Table 1, Figure 2).

Discussion. We conclude that regularization—preventing the model from perfectly fitting the training data—does matter for robust accuracy. Specifically, regularization can control the generalization gap across each group, even on the worst-case group; good robust test accuracy then becomes a question of good robust training accuracy. ERM and DRO models make different training trade-offs in the regularized regime, since no model can perfectly fit the training data; ERM models sacrifice robust for average training accuracy and therefore do poorly at test time, while DRO models maintain high robust training accuracy and therefore do well at test time. Our findings raise additional questions about the nature of generalization in neural networks, which has been predominantly studied in the context of average accuracy (Zhang et al., 2017; Hoffer et al., 2017).

3.3 Accounting for generalization through group adjustments improves DRO

We have optimized thus far for the robust *training* loss via DRO (4), relying on regularization to translate good training loss to good test loss by controlling the generalization gap δ. Here, we show that we can improve performance by directly optimizing for an estimated upper bound on the robust test loss, using ideas from structural risk minimization (Vapnik, 1992). The key consideration is that each group g has its own generalization gap $\delta_g = \mathbb{E}_{(X,Y)\sim P_g}[\ell(\theta;(X,Y))] - \mathbb{E}_{(X,Y)\sim \hat{P}_g}[\ell(\theta;(X,Y))]$. To approximate optimizing for the robust test loss $\mathcal{R}(\theta) = \max_{g\in\mathcal{G}} \mathbb{E}_{(X,Y)\sim \hat{P}_g}[\ell(\theta;(X,Y))] + \delta_g$, we propose using the simple, parameter-independent heuristic $\delta_g = C/\sqrt{n_g}$, where n_g is the group size of g and C is a model capacity constant which we treat as a hyperparameter. This gives the *group-adjusted* DRO estimator

$$\hat{\theta}_{\text{adj}} := \arg\min_{\theta\in\Theta} \max_{g\in\mathcal{G}} \mathbb{E}_{(X,Y)\sim \hat{P}_g}[\ell(\theta;(X,Y))] + \frac{C}{\sqrt{n_g}}. \tag{5}$$

The scaling with $1/\sqrt{n_g}$ reflects how smaller groups are more prone to overfitting than larger groups, and is inspired by the general size dependence of model-complexity-based generalization bounds (see, e.g., Cao et al. (2019)). Indeed, even in the regularized setting of Section 3.2, different groups g can have significant differences in their generalization gaps δ_g: for example, in the strong-weight-decay DRO model, the smallest group in Waterbirds has a train-test robust accuracy gap of 13.9% compared to just 0.6% for the largest group. By incorporating group adjustments in (5), we encourage the model to focus more on fitting the smaller groups.

Note that if we were optimizing for average loss, incorporating generalization through this penalty term would not affect the learned model because the penalty is independent of θ (i.e., the minimizer is unchanged by adding a constant to the objective). However, it matters in the DRO objective (5) because each group gets a different penalty term, and the outer $\max$ couples the groups together.

Results. We evaluate the group adjustments in the strong-weight-decay setting of Section 3.2. In Waterbirds ($\lambda = 1.0$), group adjustments improve robust test accuracy by 6.3%, cutting the error rate almost in half (Table 2 and Figure 3). The improvements in CelebA ($\lambda = 0.1$) are more modest, with robust accuracy increasing by 1.6%; weight decay is more effective in CelebA and there is not as much variation in the generalization gaps by group at $\lambda = 0.1$.

Empirically, we find that group adjustments also help in the early stopping setting of Section 3.2 (in the benchmark in the next section, we consider models with group adjustments and early stopping across a grid of weight decays, and report on the one with highest validation accuracy). However, it is difficult to rigorously study the effects of early stopping (e.g., because the group losses have not converged to a stable value), so we leave a more thorough investigation of the interaction between early stopping and group adjustments to future work.

	Average Accuracy		Robust Accuracy	
	Naïve	Adjust	Naïve	Adjust
Waterbirds	96.1	92.8	86.1	92.4
CelebA	93.5	93.4	86.7	88.3

Table 2: Average and robust accuracies with and without group adjustments. Group adjustments improve robust accuracy at a small cost in average accuracy.

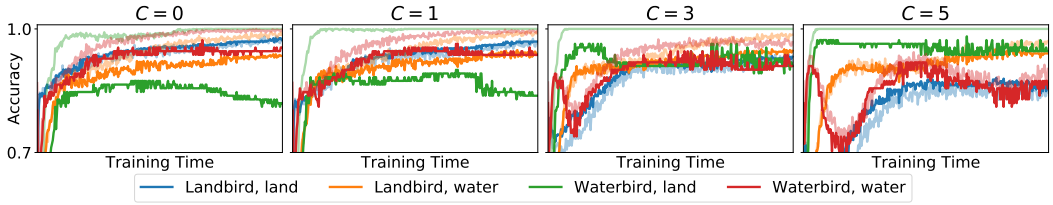

Figure 3: Training (light) and validation (dark) accuracies for each group over training progress, for different values of the adjustment constant C. When $C = 0$, the generalization gap for group 2 is large, dragging down robust accuracy; at $C = 3$ (which has the best robust validation accuracy), the accuracies are balanced; and at $C = 5$, we overcompensate for group sizes, so the smaller groups (e.g., waterbirds on land) do better at the expense of the larger groups (e.g, landbirds on land).

4 Comparison between DRO and importance weighting

We now compare DRO against importance weighting, which is frequently used in machine learning for tasks where the train and test distributions differ (Shimodaira, 2000; Byrd & Lipton, 2019), especially when weights are assigned according to low-dimensional features. Recall that in our setting, the test distribution can be any mixture of the group distributions. For some assignment of weights $w \in \Delta_m$ to groups, an importance-weighted estimator would learn

$$\hat{\theta}_w := \arg\min_{\theta \in \Theta} \mathbb{E}_{(x,y,g)\sim\hat{P}}[w_g \, \ell(\theta; (x, y))]. \tag{6}$$

While importance weighting can be a reasonable heuristic for hedging against group distribution shifts, we show that it does not reliably optimize the worst-group loss.

Empirical benchmark. We consider an importance-weighted baseline with weights set to the inverse training frequency of each group, $w_g = 1/\mathbb{E}_{g'\sim\hat{P}}[\mathbb{I}(g' = g)]$. This optimizes for a test distribution with uniform group frequencies and is analogous to the common reweighting technique for label shifts (Cui et al., 2019; Cao et al., 2019). Concretely, we train our weighted model by sampling with equal probability from each group for each minibatch (Shen et al., 2016), since a recent study found this to be more effective than similar methods (Buda et al., 2018).

Unlike group DRO, resampling for uniform group *size* does not necessarily yield uniformly low *training losses* across groups in practice, as some groups are easier to fit than others. To compare resampling (RS) with ERM and group DRO, we benchmark each objective, training models across the same grid of weight decays and early stopping at the epoch with best robust validation accuracy (Table 3).[2] In CelebA and Waterbirds, resampling performs much better than ERM but is slightly outperformed by group DRO. However, resampling fails on MultiNLI, achieving lower clean and robust accuracies than even ERM. With resampling, it appears that the rare group is overemphasized and extremely low training loss is achieved for that group at the cost of others.

Theoretical comparison. Should we expect importance weighting to learn models with good worst-case loss? We show that importance weighting and DRO can learn equivalent models in the convex setting under some importance weights, but not necessarily when the models are non-convex.

[2]To avoid advantaging the DRO models by allowing them to tune additional hyperparameters, we restrict our search for group adjustments to the single value of weight decay used in Section 3.3. See Appendix B.2.

	Average Accuracy			Robust Accuracy		
	ERM	RS	DRO	ERM	RS	DRO
Waterbirds	**97.0 (0.2)**	93.0 (0.3)	92.3 (0.3)	58.7 (2.2)	88.8 (1.4)	**91.5 (1.2)**
CelebA	**94.9 (0.2)**	92.9 (0.2)	92.9 (0.2)	47.8 (3.7)	83.3 (2.8)	**88.9 (2.3)**
MultiNLI	**82.8 (0.1)**	81.2 (0.1)	81.4 (0.1)	66.4 (1.6)	64.8 (1.6)	**77.7 (1.4)**

Table 3: Average and robust test accuracy of the model with the best validation accuracy for ERM, resampling, and group DRO, with binomial standard deviation in parenthesis. For each objective, we conduct a grid search over regularization, number of epochs, and group adjustments and report the performance of the model with highest validation accuracy.

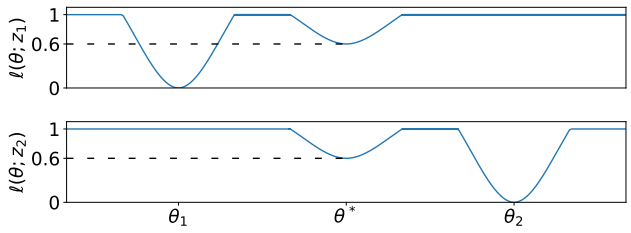

Figure 4: Toy example illustrating that DRO and importance weighting are not equivalent. The robust solution is θ^*, while any importance weighting would result in solutions at θ_1 or θ_2.

For this analysis, we shift to a more general framework of importance weighting, in which we minimize the expected loss $\mathbb{E}_{z\sim P}[w(z)\ell(\theta; z)]$ over some source distribution P. Here, we assign weights to each data point z, abstracting away from our particular choice of assigning the weights according to its group. Minimizing the weighted objective is equivalent to minimizing the expected unweighted loss $\mathbb{E}_{z\sim Q}[\ell(\theta; z)]$ over a target distribution Q such that $Q(z) \propto w(z)P(z)$.

In our setting, however, we want good worst-case performance over a family of $Q \in \mathcal{Q}$ instead of optimizing for a single Q. Do there exist weights w such that the resulting model $\hat{\theta}_w$ achieves the optimal robust risk? In the convex regime, standard arguments from convex analysis show that this is true (see Appendix A.1 for the proof):

Proposition 1. *Suppose that the loss $\ell(\cdot; z)$ is continuous and convex for all z in $\mathcal{Z}$, and the model family $\Theta \subseteq \mathbb{R}^d$ and uncertainty set $\mathcal{Q} \subseteq \mathbb{R}^m$ are convex and compact. Let θ^* be a minimizer of the robust objective $\mathcal{R}(\theta)$. Then there exists a distribution $Q^* \in \mathcal{Q}$ such that $\theta^* = \arg\min \mathbb{E}_{z\sim Q^*}[\ell(\theta; z)]$.*

In other words, any model θ^* that minimizes the robust objective also minimizes the importance-weighted objective with weights $w = Q^*/P$ for a particular choice of Q^*.

However, when the loss ℓ is non-convex—as is the case in the modern, neural-network regime—this equivalence breaks down. As a counterexample, consider a uniform data distribution P supported on just two points $\mathcal{Z} = \{z_1, z_2\}$, and let $\ell(\theta; z)$ be as in Figure 4, with $\Theta = [0, 1]$. The robust solution θ^* achieves a worst-case loss of $\mathcal{R}(\theta^*) = 0.6$. However, consider w.l.o.g. any weights $(w_1, w_2) \in \Delta_2$ with $w_1 \geq w_2$; the minimizer of the weighted loss $w_1\ell(\theta; z_1) + w_2\ell(\theta; z_2)$ is θ_1, which attains a weighted loss of ≤ 0.5 but a worst-case loss of 1.0. This negative result implies that in the non-convex setting, there may not be *any* choice of weights w that will lead to a robust model.

We note that even if there exist some importance weights that will lead to a robust model, the values of such importance weights are unknown a priori. In particular, the weights depend on θ^*, highlighting the merit of algorithms that simultaneously learn θ and $\mathcal{Q}$ like group DRO. Common choices of weights, such as inverse training frequency, are heuristics that may not yield robust solutions, as we observed in MultiNLI. As another example, we can expect these heuristics to perform poorly if we labeled backgrounds in Waterbirds in a more fine-grained fashion such that each group were of equal size: resampling would be the same as ERM, whereas DRO would still identify groups with poor performance (e.g., because their background is more unique) and upweight them.

5 ALGORITHM

To train group DRO models efficiently, we introduce an online optimization algorithm; there are no existing stochastic optimization algorithms for group DRO with convergence guarantees.

In the convex and batch case, there is a rich literature on distributionally robust optimization which treats the problem as a standard convex conic program (Ben-Tal et al., 2013; Duchi et al., 2016; Bertsimas et al., 2018; Lam & Zhou, 2015). For general non-convex DRO problems, two types of stochastic optimization methods have been proposed: (i) stochastic gradient descent (SGD) on the Lagrangian dual of the objective (Duchi & Namkoong, 2018; Hashimoto et al., 2018), and (ii) direct minimax optimization (Namkoong & Duchi, 2016). The first approach fails for group DRO because the gradient of the dual objective is difficult to estimate in a stochastic and unbiased manner.[3] An algorithm of the second type has been proposed for group DRO (Oren et al., 2019), but it lacks convergence guarantees and we observed instability in practice under some settings.

Recall that we aim to solve the optimization problem

$$\min_{\theta \in \Theta} \sup_{q \in \Delta_m} \sum_{g=1}^{m} q_g \mathbb{E}_{(X,Y)\sim P_g}[\ell(\theta; (X, Y))]. \tag{7}$$

Extending existing minimax algorithms for DRO (Namkoong & Duchi, 2016; Oren et al., 2019), we interleave gradient-based updates on θ and q. Intuitively, we maintain a distribution q over groups, with high masses on high-loss groups, and update on each example proportionally to the mass on its group. Concretely, we interleave SGD on θ and exponentiated gradient ascent on q (Algorithm 1). The key improvement from the existing group DRO algorithm (Oren et al., 2019) is that q is updated using gradients instead of picking the group with worst average loss at each iteration, which is important for algorithmic stability and obtaining convergence guarantees. In practice, we implement the algorithm with minibatching and evaluate model performance with the last iterate $\theta^{(T)}$; we observe that the algorithm reliably converges in loss.

Algorithm 1: Online optimization algorithm for group DRO

Input: Step sizes η_q, η_θ; P_g for each $g \in \mathcal{G}$
Initialize $\theta^{(0)}$ and $q^{(0)}$
for *t=1,...,T* **do**
  $g \sim \text{Uniform}(1, \ldots, m)$
  $x, y \sim P_g$
  ${q_g}^{(t)} \propto {q_g}^{(t-1)} \exp(\eta_q \ell(\theta^{(t-1)}; (x, y)))$
  $\theta^{(t)} \leftarrow \theta^{(t-1)} - \eta_\theta {q_G}^{(t)} \nabla \ell(\theta; (x, y))$
end

We analyze the convergence rate by studying the error ε_T of the average iterate $\bar{\theta}^{(1:T)}$:

$$\varepsilon_T = \max_{q \in \Delta_m} L\big(\bar{\theta}^{(1:T)}, q\big) - \min_{\theta \in \Theta} \max_{q \in \Delta_m} L\big(\theta, q\big), \tag{8}$$

where $L(\theta, q) := \sum_{g=1}^{m} q_g \mathbb{E}_{(x,y)\sim P_g}[\ell(\theta; (x, y))]$ is the expected worst-case loss. Applying results from Nemirovski & Rubinstein (2002), we can show that Algorithm 1 has a standard convergence rate of $O\big(1/\sqrt{T}\big)$ in the convex setting (proof in Section A.2):

Theorem 1. *Suppose that the loss $\ell(\cdot; (x, y))$ is non-negative, convex, B_∇-Lipschitz continuous, and bounded by B_ℓ for all (x, y) in $\mathcal{X} \times \mathcal{Y}$, and $\|\theta\|_2 \leq B_\theta$ for all $\theta \in \Theta$ with convex $\Theta \subseteq \mathbb{R}^d$. Then, the average iterate of Algorithm 1 achieves an expected error at the rate*

$$\mathbb{E}[\varepsilon_T] \leq 2m \sqrt{\frac{10[B_\theta^2 B_\nabla^2 + B_\ell^2 \log m]}{T}}. \tag{9}$$

where the expectation is taken over the randomness of the algorithm.

[3] The dual optimization problem for group DRO is $\min_{\theta,\beta} \frac{1}{\alpha} \mathbb{E}_g[\max(0, \mathbb{E}_{x,y\sim \hat{P}_g}[\ell(\theta; (x, y)) \mid g] - \beta)] + \beta$ for constant α. The max over *expected* loss makes it difficult to obtain an unbiased, stochastic gradient estimate.

6 RELATED WORK

The problem of non-uniform accuracy. Other approaches to addressing non-uniform accuracy over the data distribution include domain adaptation techniques for known target distributions (Ben-David et al., 2006; Ganin & Lempitsky, 2015) and work in ML fairness (Hardt et al., 2016; Kleinberg et al., 2017; Dwork et al., 2012). As discussed in Section 4, a classic example of the former is importance weighting (Shimodaira, 2000). Byrd & Lipton (2019) empirically study importance weighting in neural networks, demonstrating that it has little effect unless regularization is applied. This is consistent with the theoretical analysis in Wen et al. (2014), which points out that weighting has little impact in the zero-loss regime, and with our own observations in the context of DRO.

Distributionally robust optimization. Prior work has explored various definitions of the uncertainty set $\mathcal{Q}$ of possible test distributions. This is most commonly defined as a divergence ball around the training distribution over (X, Y) (Miyato et al., 2015; Esfahani & Kuhn, 2018; Ben-Tal et al., 2013; Duchi et al., 2016; Bertsimas et al., 2018; Lam & Zhou, 2015; Blanchet & Murthy, 2016). In the case of small divergence balls with radii on the order of $O(1/n)$, DRO has been used as a regularizer (Duchi & Namkoong, 2016; Shafieezadeh-Abadeh et al., 2015). Our work demonstrates that additional regularization is necessary in the large radius setting for models such as neural networks. In large radius settings, the uncertainty sets can be too pessimistic; in response, group DRO had been proposed in context of label shifts (Hu et al., 2018) and training on multiple data sources (Oren et al., 2019). DRO in general has been applied empirically, but in a different regime than ours: prior work operates in settings with training loss trade-offs and reasonable generalization, due to the low capacity of models (Duchi et al., 2019; Namkoong & Duchi, 2017), choice of conservative uncertainty set (Sinha et al., 2018), or application in generative modeling (Oren et al., 2019).

Generalization of robust models. There is extensive work investigating generalization of neural networks in terms of average loss, theoretically and empirically (Szegedy et al., 2016; Hardt et al., 2015; Hoffer et al., 2017). However, analysis on robust losses is limited. In label shifts, overfitting on rare labels has been observed and mitigative algorithms have been proposed (Buda et al., 2018; Cui et al., 2019; Cao et al., 2019). In the DRO literature, generalization bounds on the DRO objective exist for particular uncertainty sets (Duchi & Namkoong, 2018), but these bounds do not apply directly to group DRO. Invariant predictions models from the causal inference literature similarly aim to achieve high performance on a range of test distributions (Yang et al., 2019; Heinze-Deml & Meinshausen, 2017; Bühlmann & Meinshausen, 2016; Rothenhäusler et al., 2018; Peters et al., 2016). The maximin regression framework (Meinshausen & Bühlmann, 2015) also assumes group-based shifts, but focuses on settings without the generalization problems identified in our work.

7 DISCUSSION

In this paper, we analyze group DRO in the context of overparameterized, deep neural networks, highlighting the role of regularization and generalization in achieving high robust accuracy. By accounting for these factors, group DRO can significantly improve robust accuracy at only a small cost in average accuracy. The group DRO approach shows promise for preventing models from learning spurious correlations. There remain many open avenues of exploration; as an example, we could optimize for the worst α-fraction of groups instead of a single worst group, as Oren et al. (2019); Duchi et al. (2019), which would enable control over the trade-off between robust and average accuracy. More generally, our observations raise questions about the uniformity of generalization in neural networks, and suggest that future work on DRO in neural networks could have significant practical impact on applications where robust accuracy is important.

REPRODUCIBILITY

We will provide code for training group DRO models and scripts that replicate the experiments above. The constructed Waterbirds dataset will be publically available for download, with a script to adjust its generation (e.g., to choose different object backgrounds or proportions).

REFERENCES

S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira. Analysis of representations for domain adaptation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 137–144, 2006.

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

D. P. Bertsekas. *Convex Optimization Theory*. Athena Scientific Belmont, 2009.

D. Bertsimas, V. Gupta, and N. Kallus. Data-driven robust optimization. *Mathematical Programming Series A*, 167, 2018.

J. Blanchet and K. Murthy. Quantifying distributional model risk via optimal transport. *arXiv preprint arXiv:1605.01446*, 2016.

S. L. Blodgett, L. Green, and B. O'Connor. Demographic dialectal variation in social media: A case study of African-American English. In *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1119–1130, 2016.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

M. Buda, A. Maki, and M. A. Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259, 2018.

P. Bühlmann and N. Meinshausen. Magging: maximin aggregation for inhomogeneous large-scale data. In *IEEE*, 2016.

J. Buolamwini and T. Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on Fairness, Accountability and Transparency*, pp. 77–91, 2018.

J. Byrd and Z. Lipton. What is the effect of importance weighting in deep learning? In *International Conference on Machine Learning (ICML)*, pp. 872–881, 2019.

K. Cao, C. Wei, A. Gaidon, N. Arechiga, and T. Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Y. Cui, M. Jia, T. Lin, Y. Song, and S. Belongie. Class-balanced loss based on effective number of samples. In *Computer Vision and Pattern Recognition (CVPR)*, pp. 9268–9277, 2019.

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

J. Duchi and H. Namkoong. Variance-based regularization with convex objectives. *arXiv preprint arXiv:1610.02581*, 2016.

J. Duchi and H. Namkoong. Learning models with uniform performance via distributionally robust optimization. *arXiv preprint arXiv:1810.08750*, 2018.

J. Duchi, P. Glynn, and H. Namkoong. Statistics of robust optimization: A generalized empirical likelihood approach. *arXiv*, 2016.

J. C. Duchi, T. B. Hashimoto, and H. Namkoong. Distributionally robust losses against mixture covariate shifts. *preprint*, 2019.

C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel. Fairness through awareness. In *Innovations in Theoretical Computer Science (ITCS)*, pp. 214–226, 2012.

P. M. Esfahani and D. Kuhn. Data-driven distributionally robust optimization using the wasserstein metric: Performance guarantees and tractable reformulations. *Mathematical Programming*, 171 (1):115–166, 2018.

Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning (ICML)*, pp. 1180–1189, 2015.

S. Gururangan, S. Swayamdipta, O. Levy, R. Schwartz, S. R. Bowman, and N. A. Smith. Annotation artifacts in natural language inference data. *arXiv preprint arXiv:1803.02324*, 2018.

M. Hardt, B. Recht, and Y. Singer. Train faster, generalize better: Stability of stochastic gradient descent. *arXiv preprint arXiv:1509.01240*, 2015.

M. Hardt, E. Price, and N. Srebo. Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 3315–3323, 2016.

T. B. Hashimoto, M. Srivastava, H. Namkoong, and P. Liang. Fairness without demographics in repeated loss minimization. In *International Conference on Machine Learning (ICML)*, 2018.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition (CVPR)*, 2016.

C. Heinze-Deml and N. Meinshausen. Grouping-by-id: Guarding against adversarial domain shifts. *arXiv preprint arXiv:1710.11469*, 2017.

E. Hoffer, I. Hubara, and D. Soudry. Train longer, generalize better: closing the generalization gap in large batch training of neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 1731–1741, 2017.

D. Hovy and A. Sgaard. Tagging performance correlates with age. In *Association for Computational Linguistics (ACL)*, pp. 483–488, 2015.

W. Hu, G. Niu, I. Sato, and M. Sugiyama. Does distributionally robust supervised learning give robust classifiers? In *International Conference on Machine Learning (ICML)*, 2018.

S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

D. Jurgens, Y. Tsvetkov, and D. Jurafsky. Incorporating dialectal variability for socially equitable language identification. In *Association for Computational Linguistics (ACL)*, pp. 51–57, 2017.

J. Kleinberg, S. Mullainathan, and M. Raghavan. Inherent trade-offs in the fair determination of risk scores. In *Innovations in Theoretical Computer Science (ITCS)*, 2017.

H. Lam and E. Zhou. Quantifying input uncertainty in stochastic optimization. In *2015 Winter Simulation Conference*, 2015.

Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pp. 3730–3738, 2015.

A. Maurer and M. Pontil. Empirical bernstein bounds and sample variance penalization. *arXiv preprint arXiv:0907.3740*, 2009.

R. T. McCoy, E. Pavlick, and T. Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Association for Computational Linguistics (ACL)*, 2019.

N. Meinshausen and P. Bühlmann. Maximin effects in inhomogeneous large-scale data. *Annals of Statistics*, 43, 2015.

T. Miyato, S. Maeda, M. Koyama, K. Nakae, and S. Ishii. Distributional smoothing with virtual adversarial training. *arXiv*, 2015.

H. Namkoong and J. Duchi. Stochastic gradient methods for distributionally robust optimization with f-divergences. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

H. Namkoong and J. Duchi. Variance regularization with convex objectives. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

A. Nemirovski and R. Y. Rubinstein. An efficient stochastic approximation algorithm for stochastic saddle point problems. *International Series in Operations Research and Management Science*, 46:155–184, 2002.

Y. Oren, S. Sagawa, T. Hashimoto, and P. Liang. Distributionally robust language modeling. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

J. Peters, P. Bühlmann, and N. Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society. Series B (Methodological)*, 78, 2016.

A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8), 2019.

M. T. Ribeiro, S. Singh, and C. Guestrin. "why should I trust you?": Explaining the predictions of any classifier. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, 2016.

D. Rothenhäusler, P. Bühlmann, N. Meinshausen, and J. Peters. Anchor regression: heterogeneous data meets causality. *arXiv preprint arXiv:1801.06229*, 2018.

S. Shafieezadeh-Abadeh, P. M. Esfahani, and D. Kuhn. Distributionally robust logistic regression. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

L. Shen, Z. Lin, and Q. Huang. Relay backpropagation for effective learning of deep convolutional neural networks. In *European Conference on Computer Vision*, pp. 467–482, 2016.

H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90:227–244, 2000.

A. Sinha, H. Namkoong, and J. Duchi. Certifiable distributional robustness with principled adversarial training. In *International Conference on Learning Representations (ICLR)*, 2018.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research (JMLR)*, 15(1):1929–1958, 2014.

C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the Inception architecture for computer vision. In *Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2016.

R. Tatman. Gender and dialect bias in youtubes automatic captions. In *Workshop on Ethics in Natural Langauge Processing*, volume 1, pp. 53–59, 2017.

V. Vapnik. Principles of risk minimization for learning theory. In *Advances in Neural Information Processing Systems*, pp. 831–838, 1992.

C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 dataset. Technical report, California Institute of Technology, 2011.

J. Wen, C. Yu, and R. Greiner. Robust learning under uncertain test distributions: Relating covariate shift to model misspecification. In *International Conference on Machine Learning (ICML)*, pp. 631–639, 2014.

A. Williams, N. Nangia, and S. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Association for Computational Linguistics (ACL)*, pp. 1112–1122, 2018.

F. Yang, Z. Wang, and C. Heinze-Deml. Invariance-inducing regularization using worst-case transformations suffices to boost accuracy and spatial robustness. *arXiv preprint arXiv:1906.11235*, 2019.

C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations (ICLR)*, 2017.

B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6): 1452–1464, 2017.

A PROOFS

A.1 EQUIVALENCE OF DRO AND IMPORTANCE WEIGHTING IN THE CONVEX SETTING

Proposition 1. *Suppose that the loss $\ell(\theta; Z)$ is continuous and convex in θ for all Z in $\mathcal{Z}$, and the model family $\Theta \subseteq \mathbb{R}^d$ and uncertainty set $\mathcal{Q} \subseteq \mathbb{R}^m$ are convex and compact. Let θ^* be a minimizer of the robust objective $\mathcal{R}(\theta)$. Then there exists a distribution $Q^* \in \mathcal{Q}$ such that $\theta^* = \arg\min \mathbb{E}_{Z\sim Q^*}[\ell(\theta; Z)]$.*

Proof. Let $h(\theta, Q) := \mathbb{E}_{Z\sim Q}[\ell(\theta; Z)]$. Since the loss $\ell(\theta; Z)$ is continuous and convex in θ for all Z in $\mathcal{Z}$, we have that $h(\theta, Q)$ is continuous, convex in θ, and concave (linear) in Q. Moreover, since convexity and lower semi-continuity are preserved under arbitrary pointwise suprema, $\sup_{Q\in\mathcal{Q}} h(\theta, Q)$ is also convex and lower semi-continuous (therefore proper).

Together with the compactness of Θ and $\mathcal{Q}$, the above conditions imply (by Weierstrass' theorem, proposition 3.2.1, Bertsekas (2009)), that the optimal value of the DRO objective

$$\inf_{\theta\in\Theta} \mathcal{R}(\theta) = \inf_{\theta\in\Theta} \sup_{Q\in\mathcal{Q}} h(\theta, Q). \tag{10}$$

is attained at some $\theta^* \in \Theta$.

A similar argument implies that the sup-inf objective

$$\sup_{Q\in\mathcal{Q}} \inf_{\theta\in\Theta} h(\theta, Q) \tag{11}$$

attains its optimum at some $Q^* \in \mathcal{Q}$.

Moreover, because Θ and $\mathcal{Q}$ are compact and h is continuous, we have the max-min equality (see, e.g., Ex 5.25 in Boyd & Vandenberghe (2004))

$$\sup_{Q\in\mathcal{Q}} \inf_{\theta\in\Theta} h(\theta, Q) = \inf_{\theta\in\Theta} \sup_{Q\in\mathcal{Q}} h(\theta, Q). \tag{12}$$

Together, the above results imply that (θ^*, Q^*) form a saddle point (proposition 3.4.1, Bertsekas (2009)), that is,

$$\sup_{Q\in\mathcal{Q}} h(\theta^*, Q) = h(\theta^*, Q^*) = \inf_{\theta\in\Theta} h(\theta, Q^*). \tag{13}$$

In particular, the second equality indicates that the optimal DRO model θ^* also minimizes the weighted risk $h(\theta, Q^*) = \mathbb{E}_{Z\sim Q}[\ell(\theta; Z)]$, as desired.

□

A.2 CONVERGENCE RATE OF ALGORITHM 1

Theorem 1. *Suppose that the loss $\ell(\cdot; (x, y))$ is non-negative, convex, B_∇-Lipschitz continuous, and bounded by B_ℓ for all (x, y) in $\mathcal{X} \times \mathcal{Y}$, and $\|\theta\|_2 \leq B_\theta$ for all $\theta \in \Theta$ with convex $\Theta \subseteq \mathbb{R}^d$. Then, the average iterate of Algorithm 1 achieves an expected error at the rate*

$$\mathbb{E}[\varepsilon_T] \leq 2m\sqrt{\frac{10[B_\theta^2 B_\nabla^2 + B_\ell^2 \log m]}{T}}. \tag{14}$$

where the expectation is taken over the randomness of the algorithm.

Proof. Our proof is an application of the regret bound for online mirror descent on saddle point optimization from Nemirovski & Rubinstein (2002).

Consider the saddle-point optimization problem

$$\min_{\theta\in\Theta} \max_{q\in\Delta_m} \sum_{g=1}^{m} q_g f_g(\theta) \tag{15}$$

under the following assumptions:

Assumption 1. *$f_g(\theta)$ is convex on Θ.*

Assumption 2. *$f_g(\theta) = \mathbb{E}_{\xi\sim q}[F_g(\theta;\xi)]$ for some function F_g.*

Assumption 3. *We can generate i.i.d. examples $\xi \sim q$. For a given $\theta \in \Theta$ and $\xi \in \Xi$, we can compute $F_g(\theta,\xi)$ and unbiased stochastic subgradient $\nabla F_g(\theta;\xi)$, that is, $\mathbb{E}_{\xi\sim q}\left[\nabla F_g(\theta;\xi)\right] = \nabla f_g(\theta)$.*

Assume that we apply online mirror descent with some c-strongly convex norm $\|\cdot\|_\theta$ to obtain the iterates $\theta^{(1)},\dots,\theta^{(T)}$ and $q^{(1)},\dots,q^{(T)}$. From Nemirovski & Rubinstein (2002), we have the following:

Theorem 2 (Nemirovski & Rubinstein (2002), Eq 3.23)**.** *Suppose that Assumptions 1-3 hold. Then the pseudo-regret of the average iterates ${\bar{q}_g}^{(1:T)}$ and ${\bar{q}_g}^{(1:T)}$ can be bounded as*

$$\mathbb{E}\left[\max_{q\in\Delta_m}\sum_{g=1}^m q_g f_g(\bar{\theta}^{(1:T)}) - \min_{\theta\in\Theta}\sum_{g=1}^m {\bar{q}_g}^{(1:T)} f_g(\theta)\right] \leq \quad 2\sqrt{\frac{10[R_\theta^2 M_{*,\theta}^2 + M_{*,q}^2\log m]}{T}}, \tag{16}$$

where

$$\mathbb{E}\left[\left\|\nabla_\theta \sum_{g=1}^m q F_g(\theta;\xi)\right\|_{*,\theta}^2\right] \leq M_{*,\theta} \tag{17}$$

$$\mathbb{E}\left[\left\|\nabla_q \sum_{g=1}^m q F_g(\theta;\xi)\right\|_{*,q}^2\right] \leq M_{*,q} \tag{18}$$

$$R_\theta^2 = \frac{1}{c}(\max_\theta \|\theta\|_\theta^2 - \min_\theta \|\theta\|_\theta^2) \tag{19}$$

for c-strongly convex norm $\|\cdot\|_\theta$.

It remains to formulate our algorithm as an instance of online mirror descent applied to the saddle-point problem above. We start by defining the following:

Definition 1. *Let q be a distribution over $\xi = (x,y,g)$ that is a uniform mixture of individual group distributions Q_g:*

$$(x,y,g)\sim q := \frac{1}{m}\sum_{g'=1}^m Q_{g'}. \tag{20}$$

Definition 2. *Let $F_{g'}(\theta;(x,y,g))) := m\mathbb{I}[g = g']\ell(\theta;(x,y))$. Correspondingly, let $f_{g'} := \mathbb{E}_{Q_{g'}}\left[\ell(\theta;(x,y))\right]$.*

We now check that Assumptions 1-3 hold under the original assumptions in the statement of Theorem 1:

1. We assume that the loss $\ell(\cdot;(x,y))$ is non-negative, continuous, and convex for all (x,y) in $\mathcal{X}\times\mathcal{Y}$. As a result, $f_g(\theta)$ is non-negative, continuous, and convex on Θ.
2. The expected value of $F_g(\theta)$ over distribution q is $f_g(\theta)$:

$$\begin{aligned}
\mathbb{E}_{x,y,g\sim q}[F_{g'}(\theta;(x,y,g))] &= \frac{1}{m}\sum_{i=1}^m \mathbb{E}_{Q_i}\left[F_{g'}(\theta;(x,y,g)) \mid g = i\right] \\
&= \frac{1}{m}\mathbb{E}_{Q_{g'}}\left[F_{g'}(\theta;(x,y,g)) \mid g = g'\right] \\
&= \frac{1}{m}\mathbb{E}_{Q_{g'}}\left[m\ell(\theta;x,y) \mid g = g'\right] \\
&= \mathbb{E}_{Q_{g'}}\left[\ell(\theta;x,y) \mid g = g'\right] \\
&= f_{g'}(\theta).
\end{aligned}$$

3. We can compute an unbiased stochastic subgradient $\nabla F_{g'}(\theta; (x, y, g))$

$$\begin{aligned}\mathbb{E}_{x,y,g\sim q}[\nabla F_{g'}(\theta; (x, y, g))] &= \mathbb{E}_{x,y,g\sim q}[\nabla m \mathbb{I}[g = g']\ell(\theta; (x, y))] \\ &= \frac{1}{m}\sum_{i=1}^{m} \mathbb{E}_{Q_i}[\nabla m \mathbb{I}[g = g']\ell(\theta; x, y)] \\ &= \mathbb{E}_{Q_{g'}}[\nabla \ell(\theta; (x, y))] \\ &= \nabla f_g(\theta).\end{aligned}$$

Finally, we compute the constants required for the regret bound in Theorem 2. Recalling the original assumptions of Theorem 1,

1. Bounded losses: $\ell(\theta; (x, y)) \leq B_\ell$ for all x, y, θ
2. Bounded gradients: $\|\nabla \ell(\theta; (x, y))\|_2 \leq B_\nabla$ for all θ, x, y
3. Bounded parameter norm: $\|\theta\|_2 \leq B_\theta$ for all $\theta \in \Theta$,

we obtain:

$$\mathbb{E}\left[\left\|\nabla_\theta \sum_{g'=1}^{m} q_{g'} F_{g'}(\theta; (x, y, g))\right\|_{*,\theta}^2\right] \leq m^2 B_\nabla^2 = M_{*,\theta} \tag{21}$$

$$\mathbb{E}\left[\left\|\nabla_q \sum_{g'=1}^{m} q_{g'} F_{g'}(\theta; (x, y, g))\right\|_{*,q}^2\right] \leq m^2 B_\ell^2 = M_{*,q} \tag{22}$$

$$R_\theta^2 = \max_\theta \|\theta\|_\theta^2 - \min_\theta \|\theta\|_\theta^2 = B_\theta^2. \tag{23}$$

Plugging in these constants into the regret bound from Theorem 2, we obtain

$$\mathbb{E}\left[\max_{q\in\Delta_m} \sum_{g=1}^{m} q_g f_g(\bar{\theta}^{(1:T)}) - \min_{\theta\in\Theta} \sum_{g=1}^{m} \bar{q}_g{}^{(1:T)} f_g(\theta)\right] \leq \quad 2m\sqrt{\frac{10[B_\theta^2 B_\nabla^2 + B_\ell^2 \log m]}{T}} \tag{24}$$

This implies Theorem 1 because the minimax game is convex-concave. □

B Experimental details

B.1 Datasets

MultiNLI. The standard MultiNLI train-test split allocates most examples (approximately 90%) to the training set, with another 5% as a publicly-available development set and the last 5% as a held-out test set that is only accessible through online competition leaderboards (Williams et al., 2018). To accurately estimate performance on rare groups in the validation and test sets, we combine the training set and development set and then randomly resplit it to a $50 - 20 - 30$ train-val-test split that allocates more examples to the validation and test sets than the standard split.

We use the provided gold labels as the target, removing examples with no consensus gold label (as is standard procedure). We annotate an example as having a negation word if any of the words *nobody*, *no*, *never*, and *nothing* appear in the hypothesis (Gururangan et al., 2018).

Waterbirds. The CUB dataset (Wah et al., 2011) contains photographs of birds annotated by species as well as and pixel-level segmentation masks of each bird. To construct the Waterbirds dataset, we label each bird as a *waterbird* if it belongs to one of the following (groups of) species: albatross, auklet, frigatebird, fulmar, gull, jaeger, pelican, tern, fulmar, gadwall, grebe, mallard, or guillemot; and *landbird* otherwise.

To control the image background, we use the provided pixel-level segmentation masks to crop each bird out from its original background and onto a water background (categories: *ocean* or *natural lake*) or land background (categories: *bamboo forest* or *broadleaf forest*) obtained from the Places dataset (Zhou et al., 2017). In the training set, we place 90% of all waterbirds against a water background and the remaining 10% against a land background. Similarly, 90% of all landbirds are placed against a land background with the remaining 10% against water.

We refer to this combined CUB-Places dataset as the Waterbirds dataset to avoid confusion with the original fine-grained species classification task in the CUB dataset.

We use the official train-test split of the CUB dataset, randomly choosing 20% of the training data to serve as a validation set. For the validation and test sets, we allocate equal numbers of birds to each of the four groups (i.e., landbirds and waterbirds are uniformly distributed on land and water backgrounds) so as to be able to more accurately measure the performance of the rare groups. This is particularly important for the Waterbirds dataset because of its relatively small size; otherwise, the smaller groups (waterbirds on land and landbirds on water) would have too few samples to accurately estimate performance on. We note that we can only do this for the Waterbirds dataset because we control the generation process; for the other datasets, we cannot generate more samples from the rare groups.

Due to the above procedure, when reporting average test accuracy in our experiments, we calculate the average test accuracy over each group and then report a weighted average, with weights corresponding to the relative proportion of each group in the (skewed) training dataset.

CelebA. We use the official train-val-test split that accompanies the CelebA celebrity face dataset (Liu et al., 2015). We use the *Blond_Hair* attribute as the target label and the *Male* attribute as the spuriously-associated variable.

B.2 MODELS

ResNet50. We use the Pytorch `torchvision` implementation of the ResNet50 model, starting from pretrained weights.

We train the ResNet50 models using stochastic gradient descent with a momentum term of 0.9 and a batch size of 128; the original paper used batch sizes of 128 or 256 depending on the dataset (He et al., 2016). As in the original paper, we used batch normalization (Ioffe & Szegedy, 2015) and no dropout (Srivastava et al., 2014). For simplicity, we train all models without data augmentation.

We use a fixed learning rate instead of the standard adaptive learning rate schedule to make our different model types easier to directly compare, since we expected the scheduler to interact differently with different model types (e.g., due to the different definition of loss). The interaction between batch norm and weight decay means that we had to adjust learning rates for each different setting of weight decay (and each dataset). The learning rates below were chosen to be the highest learning rates that still resulted in stable optimization.

For the standard training experiments in Section 3.1, we use a weight decay of $\lambda = 0.0001$ (as in He et al. (2016)) for both Waterbirds and CelebA, with a learning rate of 0.001 for Waterbards and 0.0001 for CelebA.

For the early stopping experiments in Section 3.2, we train each ResNet50 model for 1 epoch. For the high weight decay experiments in that section, we use $\lambda = 1.0$ for Waterbirds and $\lambda = 0.1$ for CelebA, with both datasets using a learning rate of 0.00001. These settings of λ differ because we found that the lower value was sufficient for controlling overfitting on CelebA but not on Waterbirds.

For the group adjustment experiments in Section 3.3, we use the same settings of $\lambda = 1.0$ for Waterbirds and $\lambda = 0.1$ for CelebA, with both datasets using a learning rate of 0.00001. For both datasets, we search over group adjustments of $C \in \{0, 1, 2, 3, 4, 5\}$ and pick the model with the best robust validation accuracy.

For the benchmark in Section 4 (Table 3), we grid search over weight decays of $\lambda \in \{0.0001, 0.1, 1.0\}$ for Waterbirds and $\lambda \in \{0.0001, 0.01, 0.1\}$ for CelebA, using the corresponding learning rates for each weight decay and dataset listed above. (Waterbirds and CelebA at $\lambda = 0.1$, which is not listed above, both use a learning rate of 0.0001.) To avoid advantaging DRO by allow-

ing it to try many more hyperparameters, we only test group adjustments on the weight decays used in Section 3.3, i.e., $\lambda = 1.0$ for Waterbirds and $\lambda = 0.1$ for CelebA. All benchmark models were evaluated at the best early stopping epoch (as measured by robust validation accuracy).

BERT. We use the Hugging Face `pytorch-transformers` implementation of the BERT `bert-base-uncased` model, starting from pretrained weights (Devlin et al., 2018).[4] We use the default tokenizer and model settings from that implementation, including a fixed linearly-decaying learning rate starting at 0.00002, AdamW optimizer, dropout, and no weight decay, except that we use a batch size of 32 (as in Devlin et al. (2018)) instead of 8. We found that this slightly improved robust accuracy across all models and made the optimization less noisy, especially on the ERM model.

For the standard training experiments in Section 3.1, we train for 20 epochs.

For the early stopping experiments in Section 3.2, we train for 3 epochs, which is the suggested early-stopping time in Devlin et al. (2018).

For the benchmark in Section 4 (Table 3), we similarly trained for 3 epochs. All benchmark models were evaluated at the best early stopping epoch (as measured by robust validation accuracy).

[4] `https://github.com/huggingface/pytorch-transformers`