BARACK: Partially Supervised Group Robustness With Guarantees

Nimit Sohoni [1] **Maziar Sanjabi** [2] **Nicolas Ballas** [*2] **Aditya Grover** [*3] **Shaoliang Nie** [*2] **Hamed Firooz** [2]
Christopher Ré [1]

Abstract

While neural networks have shown remarkable success on classification tasks in terms of average-case performance, they often fail to perform well on certain *groups* of the data, for instance when spurious correlations are present. Unfortunately, group information may be expensive to obtain; thus, recent works in robustness and fairness have proposed ways to improve worst-group performance even when group labels are unavailable. However, these methods generally underperform methods that utilize group information at training time. In this work, we assume access to a small number of group labels alongside a larger dataset without group labels. We propose BARACK, a simple two-step framework to utilize this partial group information to improve worst-group performance: train a model to predict the missing group labels for the training data, and then use these predicted group labels in a robust optimization objective. Theoretically, we provide generalization bounds for our approach in terms of the worst-group performance, which scale with respect to both the *total* number of training points and the number of training points with *group labels*. Empirically, across four spurious correlation and robustness benchmark tasks, our method outperforms the baselines that do not use group information, even when only 1-33% of points have group labels.

1. Introduction

On classification tasks, deep neural networks can often underperform on certain groups of the data. For example, on datasets with *spurious correlations*, standard neural networks have been shown to achieve high average accuracy, yet drastically lower accuracy on groups that violate the spurious correlation (Sagawa et al., 2019). Similarly, when certain groups are underrepresented in the training data, models tend to perform poorly on these rare groups (Sohoni et al., 2020). In many settings, such as applications where fairness or safety are important, this behavior is undesirable; for example, gender classification systems have been shown to underperform for non-white faces (Buolamwini & Gebru, 2018), and medical triage systems have been shown to miss certain abnormality subtypes (Oakden-Rayner et al., 2020). To avoid this, we want to ensure *group robustness*, i.e., high accuracy on the worst-performing group.

Unfortunately, group annotations are often unavailable. Many datasets only have labels for the *task*, not groups. Group labels may also be expensive to obtain; for instance, in the common case where the group labels are finer-grained than the class labels, it may require higher annotation cost to obtain group labels than class labels (Gebru et al., 2017). This makes ensuring group robustness more challenging.

Existing works to address the issue of group robustness fall into two main categories: those that assume access to the group labels for *all* of the training data, and those that assume *no* access to the group labels for the training data. For instance, in the first category, Sagawa et al. (2019) propose *group distributionally robust optimization* (GDRO), an efficient algorithm for minimizing the worst-group loss when the groups are known. More recently, several approaches have been proposed to improve group robustness when group labels are *unavailable*. A common approach is to first estimate the group labels, then train a robust classifier using these estimated group labels (Sohoni et al., 2020; Liu et al., 2021). However, in terms of worst-group performance, the methods that require group labels unsurprisingly (and often substantially) outperform those that do not.

A fundamental question is: **can we close this gap if we have *partial* group information**? Specifically, we seek to understand the intermediate regime in which group labels are available for a (small) subset of the training data, while the remainder has class labels only. The distinction between this setting and the aforementioned prior work is akin to the difference between semi-supervised learning vs. supervised or unsupervised learning. From an application standpoint, when the identities of the groups are known, it is often

[*]Equal contribution [1]Stanford University [2]Meta AI [3]UCLA. Correspondence to: Nimit Sohoni <nims@cs.stanford.edu>.

Published at the ICML 2022 Workshop on Spurious Correlations, Invariance, and Stability. Baltimore, Maryland, USA.

feasible to obtain group labels for a *small* subset of the data. From a theoretical standpoint, the relative value of class vs. group labels for ensuring group robustness is still unknown.

To address this question, we propose BARACK,[1] a simple two-stage approach to improve group robustness for the setting wherein group labels are only known for a subset of datapoints. In the first stage of BARACK, we use the available group labels to train a model to predict the group labels on the data without group annotations. In the second stage, we use these predicted group labels in the GDRO objective (Sagawa et al., 2019) to train a robust model.

Theoretically, we show how *worst-group* performance scales with the number of total points and the number of points *with group labels*. Empirically, we show that even if only a small fraction (1-33%) of points have group labels, BARACK improves over approaches that do not use group labels.

Contributions. In summary, our main contributions are:
- We propose a simple framework, BARACK, that can improve group robustness with only a small number of group labels: we train a model to predict the missing group labels, then use these group labels in a robust training objective.
- On four benchmark tasks, we show that with only 1-33% of data having group labels, our method empirically outperforms baselines that do not use group information, approaching the performance of GDRO trained on the full dataset.
- We prove a generalization bound on the worst-group performance of our method, showing that it scales with the inverse square root of the total number of points *with group labels* in the smallest group. (In Appendix C, we show how to tighten our bound under additional assumptions.)
- We conduct ablation studies to better understand the importance of the different components of BARACK.

2. Background

2.1. Problem Setup

We consider a similar setting to Sagawa et al. (2019): we have n training points sampled IID from a distribution $\mathcal{P}$: $\{(x_i, y_i, z_i)\}_{i=1}^n \in \mathcal{X} \times \mathcal{Y} \times \mathcal{G}$. x_i denotes a datapoint, y_i its class label, and z_i its group label. However, unlike Sagawa et al. (2019), we do not assume we know all the z_i's; we only assume knowledge of $z_1, \ldots, z_m$, where $m < n$. We denote the group-labeled subset as $\mathcal{D}_1 := \{(x_i, y_i, z_i)\}_{i=1}^m$, and $\mathcal{D}_2 := \{(x_i, y_i)\}_{i=m+1}^n$ as the group-unlabeled subset. Note that we assume $\mathcal{D}_1$ and $\mathcal{D}_2$ are samples from the same distribution $\mathcal{P}$; but the z_i's are unobserved on $\mathcal{D}_2$.[2]

Our end goal is to maximize the worst-group accuracy on the task of predicting the correct *class* label for each datapoint. In other words, given a function class of classifiers $\mathcal{F}$ (where each $f \in \mathcal{F}$ is a function $\mathcal{X} \to \Delta^{|\mathcal{Y}|}$, i.e., a function that outputs probabilities for each class), we wish to find $f \in \mathcal{F}$ that maximizes $\min_{g \in \mathcal{G}} \mathbb{E}_{(x,y)|z=g}[\mathbf{1}(\text{argmax}\{f(x)\} = y)]$. In practice, we instead seek the $f \in \mathcal{F}$ that minimizes the worst-group loss over the training data:

$$\max_{g \in \mathcal{G}} \mathbb{E}_{(x_i,y_i)|z_i=g}[\ell(f(x_i), y_i)]. \quad (1)$$

When the z_i's are known, the latter problem can be solved with group DRO (GDRO) (Sagawa et al., 2019). GDRO is a stochastic optimization method designed for minimax problems of exactly the form of Eq. 1. However, in our setting, solving this problem is challenging because we only know a subset of the z_i's, so we cannot compute Eq. 1 directly.

3. Method

To address the problem of improving group robustness when only some group labels are available, we propose BARACK, a two-stage framework which leverages the group-labeled examples to generate group "pseudolabels" for the remaining datapoints, and then uses these pseudolabels to train a robust model on the target task. This two-stage approach is inspired by prior works such as JTT and GEORGE (Liu et al., 2021; Sohoni et al., 2020). However, unlike these methods which assume all group labels are unknown, BARACK is capable of exploiting the additional information in the group labels that *are* known for some datapoints. Pseudocode for BARACK's overall workflow is given in Alg. 1 (App. B).

Stage 1: Predicting group labels (via a "class-conditional" classifier). First, we train a model $f_{group} \in \mathcal{F}_1$ to *predict* the group labels for the training and validation datapoints that do not have provided group labels. To do so, we train a supervised classifier on the training points with known group labels. Despite the small number of these points, we show that this simple approach can perform surprisingly well with a key modification: we use the *class* label (assumed known for all train data) as an input to the group classifier, as the probabilities of each group can vary conditional on the class. Specifically, we compute the empirical probabilities and corresponding logits of each group conditioned on the class. For each example fed into the group classifier, these logits for its class are summed with the output of the last layer of the network. In the datasets we evaluate on, the groups are subsets of the classes, so this means that we take the softmax over the logits output by the network over all groups *belonging to the known class* to get the predicted per-group probabilities (assigning 0 probability to groups in different classes).[3] In this way, the class information helps the model learn to distinguish the groups, offsetting the dearth of data. We train f_{group} with GDRO to

[1] Name inspired by GEORGE (Sohoni et al., 2020), a baseline for the setting where no group labels are known.

[2] In other words, we assume group labels are missing at random.

[3] This can be interpreted as a form of multi-task learning (MTL) with hard weight sharing, where each class corresponds to a task.

encourage good performance at predicting each group.

To select the best group classifier model over the course of training, we use a group-labeled subset of the validation set with the same size as the group-labeled training set (ensuring that the *total* number of group labels required is small across both training and validation splits). This group classifier is then used to generate "pseudo-group-labels" for all training datapoints without a known group label. We term this approach (together with Stage 2) BARACK-Base; where unspecified, BARACK refers to BARACK-Base.

We find that this simple approach works well, and thus focus on it for the main paper. A more complex method can also be used in Stage 1: for instance, one could use semi-supervised learning to leverage the group-unlabeled points to train an improved group classifier. We term this BARACK-SSL; our preliminary experiments in Appendix D.4 and analysis in Appendix C show that this can indeed further improve performance, at the cost of added complexity and runtime.

Stage 2: Training a robust model. Intuitively, if the predicted group labels $\hat{z}_i$ from Stage 1 align well with the (unobserved) true group labels z_i, then training a model to be robust with respect to the *predicted* groups should also induce good robustness with respect to the *true* groups. Thus, we train a model $f_{robust} \in \mathcal{F}_2$ on the original task using GDRO (Sagawa et al., 2019), with the groups defined by the predicted group labels from Stage 1 (except for datapoints with known ground-truth group labels, for which we use this ground-truth). The same small group-labeled validation subset as in Stage 1 is used to validate this model.

4. Analysis

We now analyze the theoretical worst-group performance of BARACK. For brevity, much of the theory is deferred to Appendix C.

For notation, we use $\mathcal{F}_1, \mathcal{F}_2$ to denote the spaces of possible classifiers used in Stage 1 and 2 respectively. We assume $\mathcal{F}_2$ is parametrized by $\theta \in \mathbb{R}^d$, so the BARACK model $\hat{f}_{robust}$ has parameters $\hat{\theta}_{robust}$. We define $\mathcal{L}_{robust}(\theta) := \max_{g \in \mathcal{G}} \mathbb{E}_{(x,y)|z=g}[\ell(f(x;\theta), y)]$ (the worst-group loss), and $\mathcal{L}^*_{robust} := \min_{\theta \in \mathbb{R}^d} \mathcal{L}_{robust}(\theta)$. Finally, q is the population proportion of the rarest group. We assume ℓ is either the squared loss between probabilities, or the truncated cross-entropy loss, so that ℓ is bounded and Lipschitz.

First, we relate the performance of the group classifier in the first stage to the excess worst-group risk of the end model.

Theorem 4.1. *Suppose that on* each *group, the error rate of the group classifier from "Stage 1" is $\leq r$. Then with high probability, $\mathcal{L}_{robust}(\hat{\theta}_{robust}) \leq \mathcal{L}^*_{robust} + \tilde{O}\left(\frac{r}{q} + \frac{1}{\sqrt{qn}}\right)$.*

Theorem 4.1 says that the excess worst-group risk scales *linearly* in the error rate of the group classifier, plus an additional $O(\frac{1}{\sqrt{qn}})$ term which is small if the *total* number of datapoints is large. In particular, if we use standard learning-theoretic results to bound the error rate of the group classifier, then *under the assumption that the group classification problem is realizable*, we obtain the following corollary. (Realizability means there exists $f^* \in \mathcal{F}_1$ such $\mathbb{E}_{(x,y,z)\sim\mathcal{P}}[\ell(f(x,y),z)] = 0$.[4]) We show how to relax the realizability assumption in Theorem C.8 (Appendix C).

Corollary 4.2. *With high probability, for* BARACK*-Base we have $\mathcal{L}_{robust}(\hat{\theta}_{robust}) \leq \mathcal{L}^*_{robust} + \tilde{O}\left(\frac{1}{q\sqrt{m}}\right)$.*

A strength of Theorem 4.1 and Corollary 4.2 are that they do not require assumptions on the data (unlike prior work such as (Sohoni et al., 2020) which requires specific distributional assumptions to obtain generalization bounds). We discuss how to further improve our bounds in Appendix C.

5. Experiments

We empirically validate that BARACK improves group robustness on four different image classification tasks. In Section 5.2, we study how the worst-group performance of BARACK scales with the number of group labels, and compare it with several baselines. We show that with as few as 1-33% of points having group labels, BARACK attains better worst-group performance than the baselines that do not use this group information, and approaches the worst-group performance of GDRO on the full dataset as the number of group labels increases. We also show that BARACK outperforms GDRO trained on only the subset of points with group labels. In Section 5.3, we confirm that the worst-group accuracy of the final model increases with the accuracy of stage 1 of BARACK. In Appendix D, we present extensive ablations to better understand the impact of BARACK's design choices.

In our experiments, we study how performance varies as we increase the number of group-labeled examples *per group*. In other words, we pick a fixed budget of (training and validation) examples to label for each group, sampled randomly from the appropriate groups in the original dataset.

5.1. Datasets / Tasks

We evaluate on four image classification tasks: two *spurious correlation* tasks, Waterbirds and CelebA (Sagawa et al., 2019), and two tasks with a *rare group*, U-MNIST (Sohoni et al., 2020) and U-CIFAR10. Further dataset details are in Appendix D.1.

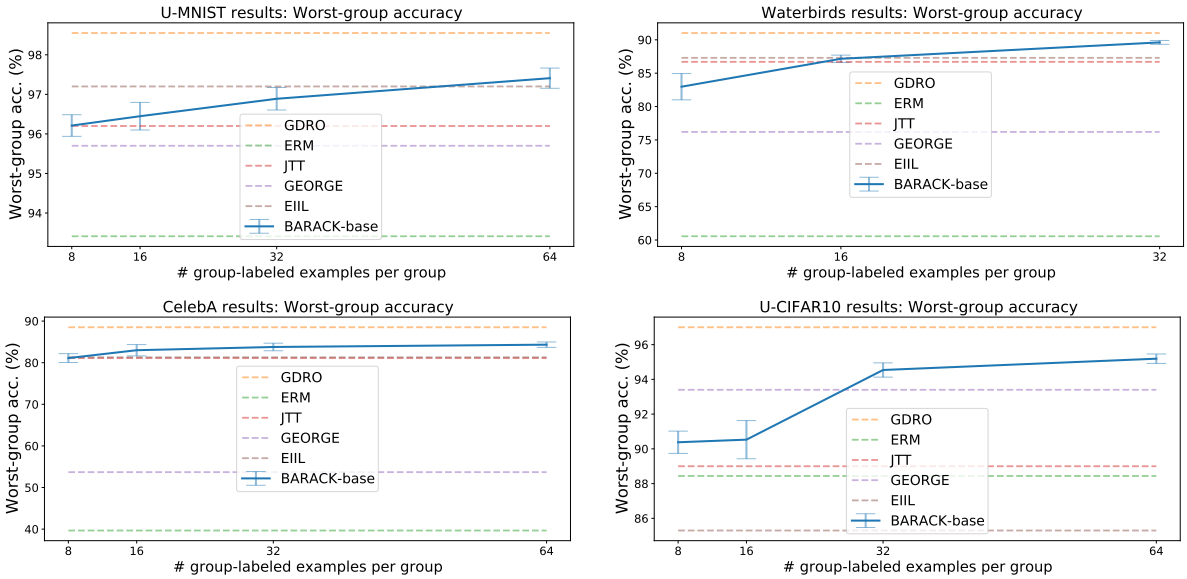

Figure 1. Main results: worst-group accuracy as a function of number of group-labeled examples. x-axis denotes the number of train examples with group labels per group (which equals the # of validation group-labeled examples per group). More results in Appendix D.

Table 1. Main results. Baselines that use no group labels for training: ERM, EIIL, GEORGE, JTT. We compare to our method (BARACK), and to GDRO run on only points with known group labels (Subset-GDRO), when there are 32 group-labeled examples from each group (in both train and validation sets). We also compare to GDRO run on the *full* dataset (Full-GDRO), which can be roughly interpreted as an "upper bound" on expected performance since it requires all group labels to be known. Additional results in Table 4 (Appendix D).

Method	U-MNIST		Waterbirds		CelebA		U-CIFAR10	
Accuracy (%)	Worst-group	Avg.	Worst-group	Avg.	Worst-group	Avg.	Worst-group	Avg.
ERM	93.4 ± 0.5	99.2 ± 0.0	60.6 ± 3.3	97.3 ± 0.1	39.7 ± 3.0	95.7 ± 0.1	88.4 ± 1.4	99.5 ± 0.1
EIIL	$\mathbf{97.2} \pm 0.5$	98.9 ± 0.2	87.3 ± 4.2	93.1 ± 0.6	81.3 ± 1.4	89.5 ± 0.4	85.3 ± 1.4	99.4 ± 0.1
GEORGE	95.7 ± 0.6	97.9 ± 0.2	76.2 ± 2.0	95.7 ± 0.5	53.7 ± 1.3	94.6 ± 0.2	93.4 ± 5.8	98.9 ± 0.3
JTT	96.2 ± 0.7	98.4 ± 0.4	88.0 ± 0.7	91.7 ± 0.8	77.8 ± 2.0	87.2 ± 1.2	89.0 ± 4.7	94.6 ± 1.3
Subset-GDRO	85.4 ± 1.4	92.4 ± 0.4	86.9 ± 1.0	88.6 ± 0.5	76.6 ± 4.4	85.5 ± 1.8	88.6 ± 2.4	95.2 ± 0.9
BARACK-base (ours)	96.9 ± 0.9	99.1 ± 0.3	$\mathbf{89.6} \pm 0.9$	94.3 ± 1.3	$\mathbf{83.8} \pm 2.7$	92.8 ± 0.6	$\mathbf{94.5} \pm 1.1$	98.9 ± 0.3
Full-GDRO	98.6 ± 0.2	99.1 ± 0.1	90.9 ± 0.2	92.8 ± 0.2	89.3 ± 0.9	92.8 ± 0.1	97.0 ± 0.3	99.2 ± 0.3

5.2. Results: Worst-Group Performance

Across the four tasks in Section 5.1, BARACK matches or improves worst-group accuracy compared to baselines that don't use group information; results are in Table 1 & Figure 1. The baselines we study are ERM, GEORGE (Sohoni et al., 2020) (which use no group information), and EIIL (Creager et al., 2021) and JTT (Liu et al., 2021) (which use group information on the val. set). Additional results are in Appendix D.

With as few as 1%-33% of training datapoints having group labels, BARACK improves over these baselines (Figure 1). For instance, CelebA has 1387 training points in the smallest group, and BARACK outperforms the baselines on CelebA with 16 group-labeled training examples per group. As the number of group-labeled points increases, the worst-group performance of BARACK approaches that of full GDRO.

[4]For standard losses such as (truncated) cross-entropy or squared loss, this implies $\text{argmax}\{f(x, y)\} = z$ w.p. 1.

In terms of *average* accuracy, BARACK and full-dataset GDRO are typically similar, while ERM is usually higher. This is expected, since on these tasks there is a trade-off between optimizing for average-case and worst-case performance, as observed in the literature (Sagawa et al., 2019). BARACK also substantially outperforms Subset-GDRO (GDRO trained only on the subset of group-labeled points) on both metrics. Subset-GDRO fails to generalize well due to the limited amount of training data it uses.

5.3. Results: Group Prediction Accuracy

In this section, we study the performance of Stage 1 of BARACK (accuracy at predicting group labels), in order to better understand the performance of Stage 2 (worst-group accuracy on the target classification task). In Table 2, we report the Stage 1 group prediction accuracies per dataset.

Table 2 shows that the group prediction models are far from perfect. Indeed, on U-CIFAR10 the accuracy at predicting the 'airplane' group is below 70%, and the average accuracy

Table 2. Group prediction accuracies for stage 1 of BARACK, for the same settings as in Table 1. Additional results in Appendix D.

Method	U-MNIST		Waterbirds		CelebA		U-CIFAR10	
	Worst-group	Avg.	Worst-group	Avg.	Worst-group	Avg.	Worst-group	Avg.
Accuracy (%)	87.7 ± 5.4	94.6 ± 1.2	91.5 ± 1.1	93.1 ± 0.8	85.9 ± 0.3	89.4 ± 0.6	67.9 ± 2.6	83.5 ± 0.6

over all groups is only $\approx 84\%$. Nevertheless, this only causes a modest drop in performance for the final robust model, as seen in Table 1.

Acknowledgments

NS performed this work during an internship with Facebook AI (now Meta AI).

References

Agarwal, A., Beygelzimer, A., Dudík, M., Langford, J., and Wallach, H. A reductions approach to fair classification. In *International Conference on Machine Learning*, pp. 60–69. PMLR, 2018.

Ahmed, F., Bengio, Y., van Seijen, H., and Courville, A. Systematic generalisation with group invariant predictions. In *International Conference on Learning Representations (ICLR)*, 2021.

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

Balashankar, A., Lees, A., Welty, C., and Subramanian, L. What is fair? Exploring Pareto-efficiency for fairness constrained classifiers. *arXiv preprint arXiv:1910.14120*, 2019.

Balcan, M.-F. and Blum, A. A discriminative model for semi-supervised learning. *Journal of the ACM (JACM)*, 2009.

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

Bhatia, K., Jain, P., and Kar, P. Robust regression via hard thresholding. *arXiv preprint arXiv:1506.02428*, 2015.

Buolamwini, J. and Gebru, T. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on Fairness, Accountability and Transparency (FAT)*, volume 81 of *Proceedings of Machine Learning Research*, pp. 77–91. PMLR, 2018. URL http://proceedings.mlr.press/v81/buolamwini18a.html.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. Unsupervised learning of visual features by contrasting cluster assignments. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, 2020.

Creager, E., Jacobsen, J.-H., and Zemel, R. Environment inference for invariant learning. In *International Conference on Machine Learning (ICML)*, 2021.

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

Gebru, T., Krause, J., Deng, J., and Fei-Fei, L. Scalable annotation of fine-grained categories without experts. In *Conference on Human Factors in Computing Systems*, 2017.

Gidaris, S., Singh, P., and Komodakis, N. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations (ICLR)*, 2018.

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

Goldstein, M., Jacobsen, J.-H., Chau, O., Saporta, A., Puli, A. M., Ranganath, R., and Miller, A. Learning invariant representations with missing data. In *Conference on Causal Learning and Reasoning (CLeaR)*, 2022.

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. Vissl. https://github.com/facebookresearch/vissl, 2021.

Hardt, M., Price, E., and Srebro, N. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29:3315–3323, 2016.

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

Huber, P. J. Robust estimation of a location parameter. In *Breakthroughs in statistics*, pp. 492–518. Springer, 1992.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Kolesnikov, A., Beyer, L., Zhai, X., Puigcerver, J., Yung, J., Gelly, S., and Houlsby, N. Big transfer (BiT): General visual representation learning. In *European Conference on Computer Vision (ECCV)*, 2020.

Krizhevsky, A. Learning multiple layers of features from tiny images. http://www.cs.toronto.edu/~kriz/cifar.html, 2009.

Lahoti, P., Beutel, A., Chen, J., Lee, K., Prost, F., Thain, N., Wang, X., and Chi, E. H. Fairness without demographics through adversarially reweighted learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. 1998.

LeCun, Y., Cortes, C., and Burges, C. MNIST handwritten digit database. *ATT Labs [Online]. http://yann.lecun.com/exdb/mnist*, 2010.

Lee, D.-H. et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICML Workshop on challenges in representation learning*, pp. 896, 2013.

Levy, D., Carmon, Y., Duchi, J. C., and Sidford, A. Large-scale methods for distributionally robust optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 8847–8860, 2020.

Li, T., Sanjabi, M., Beirami, A., and Smith, V. Fair resource allocation in federated learning. *arXiv preprint arXiv:1905.10497*, 2019.

Li, T., Beirami, A., Sanjabi, M., and Smith, V. Tilted empirical risk minimization. *arXiv preprint arXiv:2007.01162*, 2020.

Liang, P. and Ma, T. Stanford CS 229T course notes, 2019. URL http://web.stanford.edu/class/cs229t/.

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

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

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

Lokhande, V. S., Sohn, K., Yoon, J., Udell, M., Lee, C.-Y., and Pfister, T. Towards group robustness in the presence of partial group labels. *arXiv preprint arXiv:2201.03668*, 2022.

Martinez, N., Bertran, M., and Sapiro, G. Minimax pareto fairness: A multi objective perspective. In *International Conference on Machine Learning (ICML)*, 2020.

Martinez, N. L., Bertran, M. A., Papadaki, A., Rodrigues, M., and Sapiro, G. Blind Pareto fairness and subgroup robustness. In *International Conference on Machine Learning (ICML)*, 2021.

Menon, A. K., Rawat, A. S., Reddi, S. J., and Kumar, S. Can gradient clipping mitigate label noise? In *International Conference on Learning Representations*, 2019.

Mohri, M., Sivek, G., and Suresh, A. T. Agnostic federated learning. In *International Conference on Machine Learning*, pp. 4615–4625. PMLR, 2019.

Nam, J., Cha, H., Ahn, S., Lee, J., and Shin, J. Learning from failure: De-biasing classifier from biased classifier. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 20673–20684, 2020.

Nam, J., Kim, J., Lee, J., and Shin, J. Spread spurious attribute: Improving worst-group accuracy with spurious attribute estimation. In *International Conference on Learning Representations (ICLR)*, 2021.

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

Rosenberg, C., Hebert, M., and Schneiderman, H. Semi-supervised self-training of object detection models. In *IEEE Workshop on Applications of Computer Vision (WACV)*, 2005.

Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International Conference on Learning Representations*, 2019.

Sohn, K., Berthelot, D., Li, C.-L., Zhang, Z., Carlini, N., Cubuk, E. D., Kurakin, A., Zhang, H., and Raffel, C. FixMatch: Simplifying semi-supervised learning with consistency and confidence. In *NeurIPS*, 2020.

Sohoni, N., Dunnmon, J., Angus, G., Gu, A., and Ré, C. No subclass left behind: Fine-grained robustness in coarse-grained classification problems. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 19339–19352, 2020.

Utama, P. A., Moosavi, N. S., and Gurevych, I. Towards debiasing NLU models from unknown biases. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

Wang, S., Guo, W., Narasimhan, H., Cotter, A., Gupta, M., and Jordan, M. I. Robust optimization for fairness with noisy protected groups. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Xie, Q., Dai, Z., Hovy, E., Luong, M.-T., and Le, Q. V. Unsupervised data augmentation for consistency training. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020a.

Xie, Q., Luong, M.-T., Hovy, E., and Le, Q. V. Self-training with noisy student improves ImageNet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10687–10698, 2020b.

Yaghoobzadeh, Y., Mehri, S., Tachet, R., Hazen, T. J., and Sordoni, A. Increasing robustness to spurious correlations using forgettable examples. In *Conference of the European Chapter of the Association for Computational Linguistics (ACL)*, 2021.

Zhang, J., Menon, A., Veit, A., Bhojanapalli, S., Kumar, S., and Sra, S. Coping with label shift via distributionally robust optimisation. In *International Conference on Learning Representations (ICLR)*, 2021.

Zhang, M., Sohoni, N., Zhang, H., Finn, C., and Ré, C. Correct-n-contrast: a contrastive approach for improving robustness to spurious correlations. In *International Conference on Learning Representations (ICML)*, 2022.

Zoph, B., Ghiasi, G., Lin, T.-Y., Cui, Y., Liu, H., Cubuk, E. D., and Le, Q. V. Rethinking pre-training and self-training. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

A. Related Work

Beyond average performance. Our work primarily builds on prior work in the area of group robustness. This line of work has a long history in the literature; for example, see (Mohri et al., 2019; Zhang et al., 2021) and references therein. Several methods have been proposed to improve group robustness when the group labels are known at train time. In our algorithms, we focus on GDRO (Sagawa et al., 2019), a stochastic algorithm for minimizing the worst-group loss, which has been shown to substantially improve worst-group performance on tasks with spurious correlations. We use GDRO as a component of our method. While we mainly focus on worst-group accuracy, average and worst-group accuracy are not the only measures of performance of interest to ML practitioners. Several other works have used different approaches to strike a balance between important performance measures, for example through the lenses of distributional robustness (Duchi et al., 2019; Wang et al., 2020; Zhang et al., 2021; Ben-Tal et al., 2013), fairness (Hardt et al., 2016; Agarwal et al., 2018; Li et al., 2019; 2020), or outlier/noisy sample detection (Huber, 1992; Bhatia et al., 2015; Menon et al., 2019; Li et al., 2020). Balashankar et al. (2019) and Martinez et al. (2020) propose different methods for ensuring group *Pareto fairness*, which seeks to find Pareto-efficient solutions in terms of the accuracies on each group.

Group robustness without group labels. When the group labels are not known, alternative methods exist that still attempt to improve group robustness and robustness to spurious correlations. Several of these works aim to first estimate the group labels, then train a robust classifier using these estimated group labels (Sohoni et al., 2020; Nam et al., 2020; Liu et al., 2021; Zhang et al., 2022; Ahmed et al., 2021). Others make no assumptions on the structure of the groups, and simply try to perform well on "all possible" data subsets above a specified size (Levy et al., 2020; Martinez et al., 2021). Other categories of methods to improve group robustness without group labels include methods that aim to learn representations invariant to spurious features (similar to the ideas behind Invariant Risk Minimization (Arjovsky et al., 2019), but without having pre-provided environment labels), such as (Creager et al., 2021; Goldstein et al., 2022; Ahmed et al., 2021); and Pareto-fairness inspired methods, such as (Lahoti et al., 2020; Martinez et al., 2021). Unsurprisingly, all of these approaches typically underperform methods that do utilize ground-truth group labels.

Our method, BARACK, involves training two models sequentially. This is commonly used in different ways as an approach to increasing model robustness in the literature (Yaghoobzadeh et al., 2021; Utama et al., 2021). Among these works, (Liu et al., 2021; Goel et al., 2020; Creager et al., 2021; Nam et al., 2020; Sohoni et al., 2020; Zhang et al., 2022) are most relevant to our work, where a model is trained first and then the outputs of this model are used in some manner (for example, in the GDRO objective as done in (Sohoni et al., 2020)) to train the second model to be robust. Our key point of difference is that none of these works are designed to actually utilize possible group labels when they are known for some samples. BARACK can yield superior performance to these methods by utilizing such additional group information, even if it is limited.

Semi-supervised learning. Semi-supervised learning (SSL) is a rich field with several recent developments. In the appendix, we explore the use of FixMatch (Sohn et al., 2020), a recent state-of-the-art method for SSL, for group classification. Other recent successful approaches to semi-supervised learning involve learning *self-supervised* representations (without using labels), and then using the labeled examples for fine-tuning; examples include (Xie et al., 2020a; Chen et al., 2020; Caron et al., 2020).

Our work also has connections to self-training. In standard self-training, a labeled dataset is used to train a model to generate pseudolabels for a separate unlabeled dataset; the labeled and pseudolabeled data are then used together to train a downstream model (Zoph et al., 2020; Xie et al., 2020b; Lee et al., 2013; Rosenberg et al., 2005). In our work, we instead generate pseudolabels for the task of classifying the *groups*, which are then used to train a robust model for the original task.

Concurrent work. We would also like to acknowledge the following important concurrent work: Spread Spurious Attribute (SSA) (Nam et al., 2021), which considers a very similar problem to ours: specifically, they consider the problem of group robustness when there are spurious attributes that are known for a subset of the training data. Their proposed algorithm can be viewed as a special case of our general two-step framework (BARACK), in which they use semi-supervised learning techniques for the group prediction stage. Compared to (Nam et al., 2021), our work is more focused on understanding and analyzing the effectiveness of BARACK through theoretical and ablation analyses. We believe that these analyses provide valuable insight regardless of the precise method used to estimate group labels. As we show in our experiments and analysis, while the basic version of BARACK performs quite well, better training methods for the first (group classification) stage, such as with SSL, can translate to better results (possibly at the cost of increased computational complexity).

(Lokhande et al., 2022) also address a similar version of our partial group robustness problem, using a different approach based on minimizing an upper bound to the GDRO loss. However, unlike our work (and that of (Nam et al., 2021)), they avoid any estimation of missing group labels due to privacy considerations, which unfortunately results in substantially lower worst-group performance (albeit still better than ERM) due to their upper bound function possibly being quite loose.

B. Algorithm Pseudocodes

On the following pages, we include algorithm boxes for BARACK, as well as the subroutines used therein. We also provide a schematic illustrating BARACK (Figure 7) for explanatory purposes.

Additionally, to demonstrate the flexibility of our BARACK framework, we present Algorithm 5 (BARACK-SSL), an instantiation of the general BARACK framework (Algorithm 1) which uses semi-supervised learning to train the group classifier. Our preliminary empirical results on BARACK-SSL (Appendix D.4) and theoretical analysis of BARACK-SSL (Appendix C.10) suggest that BARACK-SSL is a promising approach to *further* improve performance in our setting.

Algorithm 1 BARACK (General)

Require: Group-labeled data $\mathcal{D}_1 = \{(x_i, y_i, z_i)\}_{i=1}^m$, group-unlabeled data $\mathcal{D}_2 = \{(x_i, y_i)\}_{i=m+1}^n$.

1: $\hat{f}_{group} \leftarrow \text{TRAIN}(\mathcal{D}_1, \mathcal{D}_2)$
2: $(\hat{z}_{m+1}, \ldots, \hat{z}_n) \leftarrow \text{PREDICT}(\hat{f}_{group}, \mathcal{D}_2)$
3: $(\hat{z}_1, \ldots, \hat{z}_m) \leftarrow (z_1, \ldots, z_m)$
4: $\hat{f}_{robust} \leftarrow \text{TRAIN\_ROBUST}(\{(x_i, y_i, \hat{z}_i)\}_{i=1}^n)$

output : Final model $\hat{f}_{robust}$.

Figure 2. General overview of BARACK, our algorithmic framework. BARACK can be instantiated with specific choices of methods to train the initial group classifier and the final robust model (for example, see BARACK-Base, Algorithm 2).

Algorithm 2 BARACK-Base

Require: Group-labeled data $\mathcal{D}_1 = \{(x_i, y_i, z_i)\}_{i=1}^m$, group-unlabeled data $\mathcal{D}_2 = \{(x_i, y_i)\}_{i=m+1}^n$.

1: $\hat{f}_{group} \leftarrow \text{TRAIN\_SUPERVISED}(\mathcal{D}_1)$
2: $(\hat{z}_{m+1}, \ldots, \hat{z}_n) \leftarrow \text{PREDICT}(\hat{f}_{group}, \mathcal{D}_2)$
3: $(\hat{z}_1, \ldots, \hat{z}_m) \leftarrow (z_1, \ldots, z_m)$
4: $\hat{f}_{robust} \leftarrow \text{GDRO}(\{(x_i, y_i, \hat{z}_i)\}_{i=1}^n)$

output : Final model $\hat{f}_{robust}$.

Figure 3. The base version of BARACK, used in most experiments. A supervised classifier is trained on the group-labeled examples and generates pseudolabels for the remaining training data. The pseudolabels are used to train a more robust model with GDRO. Pseudocode for subroutines is below.

Algorithm 3 TRAIN_SUPERVISED

Require: Group-labeled data $\mathcal{D}_1 = \{(x_i, y_i, z_i)\}_{i=1}^m$, group-labeled validation data $\mathcal{D}_{val}$
Initialize model $f_\theta \in \mathcal{F}_1$
Initialize acc_best = 0
for epoch in $1, \ldots, T$ **do**
  for i in $1, \ldots, m$ **do**
    $\theta \leftarrow$ GDRO_Update$(\ell, f_\theta(x_i, y_i), z_i)$
  end for
  if ValidationSubsetWorstGroupAcc$(f_\theta) >$ acc_best **then**
    $\hat{f}_{group} \leftarrow f_\theta$
    acc_best $\leftarrow$ ValidationSubsetWorstGroupAcc(f_θ)
  end if
end for
output : Group prediction model $\hat{f}_{group}$.

Figure 4. Details for training the supervised *group* classifier. In our experiments, it is typically initialized from a pretrained ResNet-50 model, except for MNIST where it is a randomly initialized LeNet-5. We minimize the GDRO training loss (in our experiments, ℓ is the cross-entropy loss), and select the model that does best on the validation subset with group labels, in terms of the worst-group accuracy on that subset. The group classifier takes both the features x and class label y as input. (For the specific GDRO update equation, see (Sagawa et al., 2019). In practice, we do a minibatched version of the above.)

Algorithm 4 PREDICT

Require: Group prediction model $\hat{f}_{group}$, group-unlabeled training data $\mathcal{D}_2 = \{(x_i, y_i)\}_{i=m+1}^n$.
for $i = 1, \ldots, m$ **do**
  $\hat{z}_i \leftarrow \text{argmax}_{g \in \mathcal{G}}\, \hat{f}_{group}(x_i, y_i)$
end for
output : $\{\hat{z}_i\}_{i=1}^m$

Figure 5. Details for extracting group predictions from the trained group prediction model.

Algorithm 5 BARACK-SSL

Require: Group-labeled data $\mathcal{D}_1 = \{(x_i, y_i, z_i)\}_{i=1}^m$, group-unlabeled data $\mathcal{D}_2 = \{(x_i, y_i)\}_{i=m+1}^n$.
$\hat{f}_{group} \leftarrow$ TRAIN_FIXMATCH$(\mathcal{D}_1, \mathcal{D}_2)$
$(\hat{z}_{m+1}, \ldots, \hat{z}_n) \leftarrow$ PREDICT$(\hat{f}_{group}, \mathcal{D}_2)$
$(\hat{z}_1, \ldots, \hat{z}_m) \leftarrow (z_1, \ldots, z_m)$
$\hat{f}_{robust} \leftarrow$ GDRO$(\{(x_i, y_i, \hat{z}_i)\}_{i=1}^n)$
output : Final model $\hat{f}_{robust}$.

Figure 6. BARACK-SSL. A semi-supervised model is trained on both the group-labeled training examples and the training examples with only class labels, and generates group pseudolabels for the training data without group labels. These pseudolabels are used to train a more robust model using GDRO (Sagawa et al., 2019). For FixMatch training, we use the standard FixMatch algorithm (Sohn et al., 2020) except we modify it to take the class label y as input as well, similar to how we train BARACK-BASE.

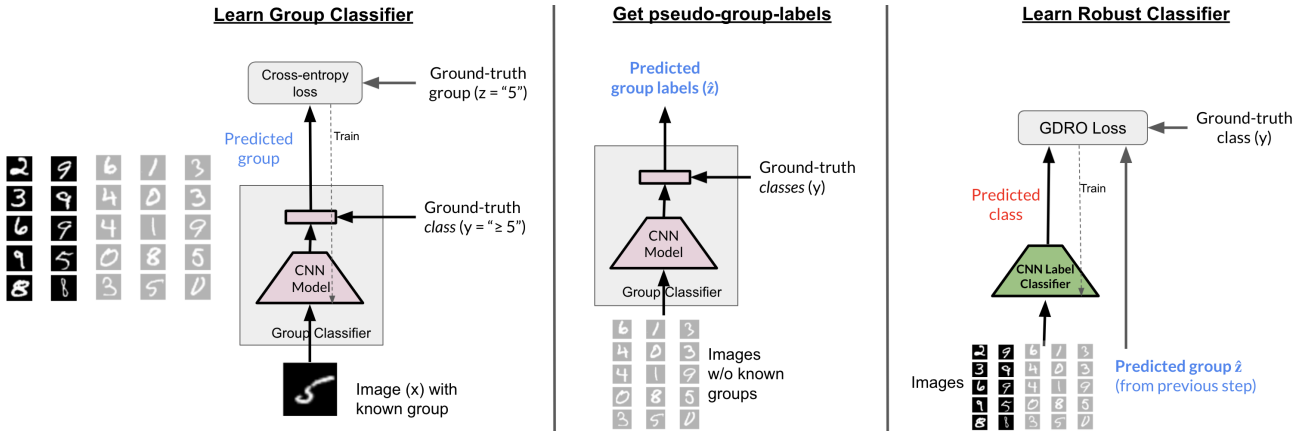

Figure 7. Schematic of BARACK-Base. Inputs are images & class labels; some images also have known group labels (darker examples). We first train a model to classify groups using the images with known group labels (left). Then, we use this model to compute "group pseudolabels" for the remaining training images (middle). Finally, we use these pseudolabels in GDRO to train a robust model (right).

C. Theoretical Analysis and Proofs

C.1. Overview

First, we state an upper bound on the worst-group loss of GDRO, and a *lower* bound on that of ERM. Next, we restate a generalization bound on the worst-group loss of BARACK (Theorem 4.1), and discuss how this result relates to the GDRO and ERM bounds. Proofs are provided in the following subsections.

First, we show that if GDRO is trained on the dataset of the m group-labeled points, the excess worst-group risk of the resulting model (compared to the worst-group-optimal model, i.e. the model in $\mathcal{F}_2$ with the lowest population worst-group performance on the task) scales as $\tilde{O}(1/\sqrt{qm})$.

Lemma C.1. *Let $\hat{f}_{gdro,labeled} \in \mathcal{F}_2$ (with associated parameters $\hat{\theta}_{gdro,labeled}$) denote the GDRO classifier trained on $\mathcal{D}_1$ only. Then with high probability,*
$\mathcal{L}_{robust}(\hat{\theta}_{gdro,labeled}) \leq \mathcal{L}^*_{robust} + \tilde{O}(\frac{1}{\sqrt{qm}})$.

By contrast, ERM (even trained on the full dataset) can result in a worst-group risk multiple times higher than that of the optimal worst-group model:

Lemma C.2. *Let $\hat{f}_{erm,full} \in \mathcal{F}_2$ (with associated parameters $\hat{\theta}_{erm,full}$) denote the ERM classifier trained on the full dataset. There exists a distribution $\mathcal{P}$ such that with high probability, $\mathcal{L}_{robust}(\hat{\theta}_{erm,full}) \geq \frac{\log(1/q)}{\log(|\mathcal{Y}|)}\mathcal{L}^*_{robust} - \tilde{O}(1/\sqrt{n})$.*

For comparison, recall our bound (Theorem 4.1) on the generalization risk of BARACK based on the group classifier accuracy, and the resulting Corollary 4.2 under the assumption of realizability of the group classification problem:

Theorem 4.1. *Suppose that on* each *group, the error rate of the group classifier from "Stage 1" is $\leq r$. Then with high probability, $\mathcal{L}_{robust}(\hat{\theta}_{robust}) \leq \mathcal{L}^*_{robust} + \tilde{O}\left(\frac{r}{q} + \frac{1}{\sqrt{qn}}\right)$.*

Corollary C.3. *With high probability, for* BARACK*-Base we have $\mathcal{L}_{robust}(\hat{\theta}_{robust}) \leq \mathcal{L}^*_{robust} + \tilde{O}\left(\frac{1}{q\sqrt{m}}\right)$.*

(Note: Realizability of the group classifier is not a fundamentally necessary assumption, and can be relaxed; in Appendix C.6 we provide Theorem C.8, an analogue to Theorem 4.1 with an improved bound in the non-realizable case. For ease of exposition, we defer further description of Theorem C.8 to Appendix C.6.)

As mentioned, Theorem 4.1 and Corollary 4.2 do not require data distribution assumptions. However, the downside of Theorem 4.1 is that the bound is relatively weak unless the group classifier is known to perform near-optimally; indeed, Corollary 4.2 yields a slightly weaker asymptotic bound than GDRO on the labeled data alone. This result can be improved when further assumptions are made to guarantee a stronger bound on the group classification error.

For instance, in Corollary C.13 (Appendix C.10), we show that the excess worst-group generalization error rate bound can be improved to $O(1/\sqrt{n})$ when *semi-supervised* learning is used to train the group classifier. With an appropriate choice of semi-supervised learning method (such as FixMatch (Sohn et al., 2020)), if $m = \Omega(\sqrt{n})$, then under appropriate conditions, the worst-group generalization error rate bound of BARACK-SSL is $O(1/\sqrt{n})$. This result is based on the PAC-learning based results of (Balcan & Blum, 2009), and requires the assumptions therein as well as realizability (the exact conditions are somewhat technical, and are deferred to the discussion in Appendix C.10). This translates to an excess robust risk of $O(1/\sqrt{n})$ for the final model, as stated in Corollary C.13.

Next, another natural question is how the performance of BARACK-Base compares to that of ERM. We study this in Corollary C.4.

Corollary C.4. *In addition to the assumptions of Theorem 4.1, suppose that for all $f \in \mathcal{F}_2$, $\ell(f(x), y) \perp \text{argmax}\{\hat{f}_{group}(x, y)\} \mid z$. Let θ^*_{avg} be the minimizer of the population* average *loss $\mathbb{E}_{(x,y)\sim P}[\ell(x, y; \theta)]$. Then for* BARACK*-Base we have $\mathcal{L}_{robust}(\hat{\theta}_{robust}) \leq \min\left(\mathcal{L}^*_{robust} + \tilde{O}(\frac{1}{q\sqrt{m}}), \mathcal{L}_{robust}(\theta^*_{avg}) + \tilde{O}(\frac{1}{\sqrt{qn}})\right)$ w.h.p.*

For comparison, the robust loss of the ERM model is upper bounded by $\mathcal{L}_{robust}(\theta^*_{avg}) + \tilde{O}(\frac{1}{\sqrt{qn}})$ with high probability.

Corollary C.4 says that if we can assume that the errors made by f_{group} are "random" conditioned on the true group identity—i.e., they do not affect the distribution of the loss on the target task—then in addition to the bound of Theorem 4.1, we can also guarantee that the worst-group loss is at least as good as that of the ERM model (plus small $\tilde{O}(1/\sqrt{qn})$ noise).

Of course, this "random error" assumption is very strong; nevertheless, in Appendix D.3.1 we compare the performance of BARACK-Base and simulated "group predictions" with the same confusion matrix as those of BARACK-Base but random errors, and find that the BARACK predictions do not substantially degrade performance compared to these randomized predictions. Thus, we hypothesize that the errors made by BARACK-Base are "sufficiently random" to make the conditions of Corollary C.4 hold approximately.

We now proceed to prove the results stated in this section.

C.2. Notation

The notation "$i \in [b]$" means $i \in \{1, 2, \ldots, b\}$. $\log$ denotes the natural logarithm. Where unspecified, $\|\cdot\|$ denotes the Euclidean norm $\|\cdot\|_2$. Δ^d denotes the d-coordinate ($d-1$ dimensional) simplex.

We have a training dataset $\{(x_i, y_i, z_i)\}_{i=1}^n \in \mathcal{X} \times \mathcal{Y} \times \mathcal{G}$. Here x_i is a datapoint, y_i is a discrete class label, and z_i is a discrete group label. Let $\mathcal{F}_1$ be a function class $\{f_\theta : \mathcal{X} \times \mathcal{Y} \to \Delta^{|\mathcal{G}|}\}$, where each f_θ is a member of $\mathcal{F}_1$ parameterized by the vector $\theta \in \mathbb{R}^d$. (The inputs to f_θ are the features x and class label y, and the output is a vector of predicted probabilities for each group.) Define $\mathcal{F}_1^R := \{f_\theta \in \mathcal{F}_1 : \|\theta\|_2 \leq R\}$. Similarly let $\mathcal{F}_2$ be a function class $\{f_\theta : \mathcal{X} \to \Delta^{|\mathcal{Y}|}\}$ and $\mathcal{F}_2^R := \{f_\theta \in \mathcal{F}_2 : \|\theta\|_2 \leq R\}$. Let $\ell(\cdot, \cdot)$ be a nonnegative loss function globally bounded by B; ℓ is either in $\mathbb{R}^{|\mathcal{Y}|} \times \mathcal{Y} \to \mathbb{R}$ or $\mathbb{R}^{|\mathcal{G}|} \times \mathcal{G} \to \mathbb{R}$ depending on whether it is for the group classification task or target classification task, and will be clear from context. We will use the notations $\ell(f_\theta(x), y)$ and $\ell(x, y; \theta)$ interchangeably (or even $\ell(f(x), y)$ or $\ell(x, y)$ when clear from context).

We assume that the datapoints are IID samples from a distribution P, i.e., $(x_i, y_i, z_i) \sim P$. We can write P as a mixture of distributions $p_1, \ldots, p_{|\mathcal{G}|}$, where p_g is the distribution conditioned on the group label being equal to g. By overloading notation, we will also write $\mathbb{E}_{(x,y)\sim p_g}$ to denote the expectation of a quantity conditioned on the group label being g (i.e., we will not write $(x, y, z) \sim p_g$ because $z = g$ always for points sampled from p_g). Let $q_1, \ldots, q_{|\mathcal{G}|}$ be the corresponding mixture weights, i.e., $q_g = P(z = g)$, and let $q := \min_{g \in \mathcal{G}} q_g$.

Recall that we denote $\mathcal{D}_1 := \{(x_i, y_i, z_i)\}_{i=1}^m$ to be the group-labeled dataset of m points, and $\mathcal{D}_2 := \{(x_i, y_i)\}_{i=m+1}^n$ to be the dataset of $n - m$ group-unlabeled points. We are interested in the regime where m is small compared to n, so we shall implicitly assume that $n - m$ is $\Omega(n)$. In our theoretical results we assume for simplicity that $\mathcal{D}_1$ and $\mathcal{D}_2$ are samples from the same distribution (although the z_i's are unobserved on $\mathcal{D}_2$). However, our analysis easily extends to the case where $\mathcal{D}_1, \mathcal{D}_2$ are sampled from distributions on which the group proportions $q_1, \ldots, q_{|\mathcal{G}|}$ differ, but the per-group distributions $p_g = P(x, y | z = g)$ are the same. (This only complicates notation, as we need to distinguish the different $q = \min_{g \in \mathcal{G}} q_g$ between $\mathcal{D}_1$ and $\mathcal{D}_2$.)

We will use the notation $\mathcal{L}_{robust}(\theta)$ to denote either $\max_{g \in \mathcal{G}} \mathbb{E}[\ell((x, y), z; \theta) | z = g]$ (worst-group population loss for the task of classifying the *groups*) or
$\max_{g \in \mathcal{G}} \mathbb{E}[\ell(x, y; \theta) | z = g]$ (worst-group population loss for the task of classifying the *classes*), which will be clear in context. We will also use $\mathcal{L}^*_{robust}$ as shorthand for $\min_\theta \mathcal{L}_{robust}(\theta)$.

C.3. "Helper" Results

We will use the following standard result from learning theory (Liang & Ma, 2019):

Theorem C.5. *Suppose $\ell(x, y; \theta)$ is nonnegative, globally bounded by B and L-Lipschitz continuous. Define $L(\theta) := \mathbb{E}_{(x,y)\sim P}[\ell(x, y; \theta)]$ and $\hat{L}(\theta) := \frac{1}{n}\sum_{i=1}^n \ell(x_i, y_i; \theta)$, where $\{(x_i, y_i)\}_{i=1}^n$ are sampled IID from distribution P. Let $p > 0$. Then, with probability $\geq 1 - O(e^{-p})$, for all θ such that $\|\theta\|_2 \leq R$ we have $|\hat{L}(\theta) - L(\theta)| \leq O\left(B\sqrt{\frac{p \max(\log(LRn), 1)}{n}}\right)$.*

We will also use the following simple lemma relating the minimizer of a "perturbed" GDRO-style objective to the minimizer of the unperturbed version.

Lemma C.6. *Define $f(\theta) = \max_{k \in [G]} f_k(\theta)$ and $\tilde{f}(\theta) = \max_{k \in [G]} (f_k(\theta) + \epsilon_k)$ where $|\epsilon_k| \leq \epsilon$. Let the minimizers of $f, \tilde{f}$ be $\theta^*, \tilde{\theta}^*$ respectively. Then $f(\tilde{\theta}^*) - f(\theta^*) \leq 2\epsilon$.*

Proof. Note that $\tilde{f}(\theta^*) \leq f(\theta^*) + \epsilon$. Similarly $f(\tilde{\theta}^*) \leq \tilde{f}(\tilde{\theta}^*) + \epsilon$. So $f(\tilde{\theta}^*) \leq \tilde{f}(\tilde{\theta}^*) + \epsilon \leq \tilde{f}(\theta^*) + \epsilon \leq f(\theta^*) + 2\epsilon$. □

C.4. Warm-up: Proof of Group DRO Generalization Bound (Lemma C.1)

We restate Lemma C.1 below:

Lemma C.7. *Let $\hat{f}_{gdro,labeled} \in \mathcal{F}_2$ (with associated parameters $\hat{\theta}_{gdro,labeled}$) denote the GDRO classifier trained on $\mathcal{D}_1$ only. Then with high probability,*
$\mathcal{L}_{robust}(\hat{\theta}_{gdro,labeled}) \leq \mathcal{L}^*_{robust} + \tilde{O}(\frac{1}{\sqrt{qm}})$.

Proof. The population group DRO loss is $\mathcal{L}_{robust}(\theta) := \max_{g \in \mathcal{G}} \left(\mathbb{E}_{(x,y)\sim p_g}[\ell(x, y; \theta)]\right)$, i.e. the maximum of the average per-group losses. Here p_g denotes the conditional distribution $P(x, y|z = g)$. For $g \in \mathcal{G}$, denote S_g to be the subset of points on $\mathcal{D}_1$ such that the group label is g. The empirical GDRO loss on $\mathcal{D}_1$ is $\mathcal{L}_{robust,\mathcal{D}_1}(\theta) := \max_{g \in [G]} \left(\mathbb{E}_{(x,y)\sim S_g}[\ell(x, y; \theta)]\right)$. Note that each of the per-group losses is simply the empirical estimate of the corresponding population per-group loss, over the set S_g.

We can apply Theorem C.5 to each of the groups individually, since each set S_g is an IID sample from p_g. Thus, we obtain that for a given group g, $\left|\mathbb{E}_{(x,y)\sim S_g}[\ell(x, y; \theta)] - \mathbb{E}_{(x,y)\sim p_g}[\ell(x, y; \theta)]\right| \leq O(\sqrt{\frac{p \log |S_g|}{|S_g|}})$ for *all* θ such that $\|\theta\|_2 \leq R$, with probability $\geq 1 - O(e^{-p})$. ($|S_g|$ is the size of set S_g.) Thus by union bound, $\max_{g \in \mathcal{G}} \left|\mathbb{E}_{(x,y)\sim S_g}[\ell(x, y; \theta)] - \mathbb{E}_{(x,y)\sim p_g}[\ell(x, y; \theta)]\right| \leq O\left(\max_{S_g} \sqrt{\frac{p \log |S_g|}{|S_g|}}\right)$ w.p. $\geq 1 - O(|\mathcal{G}|e^{-p})$. (Here and henceforth we omit the constants B, L, R from the notation.)

Let $q_g := P(z = g)$, i.e., the population fraction of group g. By Hoeffding's inequality, for a given group g we have $|S_g - q_g m| \leq \sqrt{m \log m}$ with probability $\geq 1 - 2/m^2$. Thus by union bound and the fact that $q_g \geq q$, we have

$$\min_{g \in \mathcal{G}} |S_g| \geq qm - \sqrt{m \log m} = (1 - \tfrac{\log m}{q\sqrt{m}})qm$$

with probability greater than $1 - 2|\mathcal{G}|/m^2$. Thus $\max_{g \in \mathcal{G}} \frac{\log |S_g|}{|S_g|} \leq \frac{\log m}{qm} \cdot \frac{1}{1 - \frac{\log m}{q\sqrt{m}}}$ with probability greater than $1 - 2|\mathcal{G}|/m^2$.

As $\frac{1}{1-x} \leq 1 + 2x$ for all $x \in [0, 1/2]$, we thus have

$$\max_{g \in \mathcal{G}} \frac{\log |S_g|}{|S_g|} \leq \min\left\{1, \frac{\log m}{qm} + 2\frac{\log m}{qm} \cdot \frac{\log m}{q\sqrt{m}}\right\}$$

for sufficiently large m (relative to q). Thus

$$\max_g \sqrt{\frac{p \log |S_g|}{|S_g|}} = O\left(\sqrt{\tfrac{\log m}{qm}}\right)$$

with probability greater than $1 - 2|\mathcal{G}|/m^2$.

To recap, the training loss is $\max_{g \in \mathcal{G}} \mathbb{E}_{(x,y)\sim S_g} \mathbb{E}[\ell(x, y; \theta)]$ and we showed that

$$|\mathbb{E}_{(x,y)\sim S_g} \mathbb{E}[\ell(x, y; \theta)] - \mathbb{E}_{(x,y)\sim p_g} \mathbb{E}[\ell(x, y; \theta)]| \leq O\left(\sqrt{\tfrac{\log m}{qm}}\right)$$

for all $g \in \mathcal{G}$ and all θ with $\|\theta\|_2 \leq R$ with high probability, so applying Lemma C.6 and union bound, we have that

$$\mathcal{L}_{robust}(\hat{\theta}) - \mathcal{L}^*_{robust} = |\mathcal{L}_{robust}(\hat{\theta}) - \mathcal{L}^*_{robust}| \leq O\left(\sqrt{\tfrac{\log m}{qm}}\right) = \tilde{O}(\tfrac{1}{\sqrt{qm}})$$

with high probability, as desired. Note that as long as we optimize over a bounded domain that is sufficiently large to contain θ^*, there exists a valid norm constraint R such that all θ under consideration have $\|\theta\|_2 \leq R$.

C.4.1. RELATING LOSS TO CLASSIFICATION ERROR

For example, we could let ℓ be the truncated cross-entropy loss (i.e., the cross-entropy loss clipped to remain in $[0, B]$ for some large constant B in order to ensure boundedness). Observe that the classification error is upper bounded by the cross-entropy loss divided by $\log(|\mathcal{Y}|)$, where $|\mathcal{Y}|$ is the number of classes: as the class prediction is the class with highest predicted probability, if the model makes an error then the predicted probability of the correct class is at most $1/|\mathcal{Y}|$, which means that the cross-entropy loss for that example is $\geq -\log(1/\mathcal{Y}) = \log|\mathcal{Y}|$. In other words, if $\text{argmax}\{f(x)\} \neq y$ then $\ell(f(x), y) \geq \log|\mathcal{Y}|$. Thus $\mathbf{1}(\text{argmax}\{f(x)\} \neq y) \leq \frac{\ell(f(x),y)}{\log|\mathcal{Y}|}$, and so $\mathbb{E}[\mathbf{1}(\text{argmax}\{f(x)\} \neq y)] \leq \frac{1}{\log|\mathcal{Y}|}\mathbb{E}[\ell(f(x), y)]$. (Thus, the clipping constant B just needs to be $\geq \log|\mathcal{Y}|$.)

We could also let ℓ be the squared loss (the square of 1 minus the predicted probability of the correct class), which is 2-Lipschitz and bounded by 1. The classification error is upper bounded by $\frac{|\mathcal{Y}|}{|\mathcal{Y}|-1}$ times the squared loss (since when the model makes an error, the loss on that example must be at least $(1-1/|\mathcal{Y}|)^2+(|\mathcal{Y}|-1)/|\mathcal{Y}|^2 = 1-2/|\mathcal{Y}|+1/|\mathcal{Y}| = \frac{|\mathcal{Y}|-1}{|\mathcal{Y}|}$).

C.5. Proof of Theorem 4.1

We restate Theorem 4.1 below, and prove it in the following two subsubsections.

Theorem 4.1. *Suppose that on* each *group, the error rate of the group classifier from "Stage 1" is* $\leq r$. *Then with high probability,* $\mathcal{L}_{robust}(\hat{\theta}_{robust}) \leq \mathcal{L}^*_{robust} + \tilde{O}\left(\frac{r}{q} + \frac{1}{\sqrt{qn}}\right)$.

C.5.1. PER-GROUP POPULATION LOSS VS. PER-ESTIMATED-GROUP POPULATION LOSS.

Let $\tilde{z}$ denote the prediction of the Stage 1 group classifier, i.e. $\tilde{z} = \text{argmax}\{\hat{f}_{group}(x, y)\}$. The assumption on the error rate on each group is equivalent to assuming $P(\tilde{z} \neq z|z = g) \leq r$ for each $g \in \mathcal{G}$. Notice that $P(z \neq \tilde{z}|\tilde{z} = g) = \sum_{g' \in |\mathcal{G}|, g' \neq g} P(z = g'|\tilde{z} = g) = \sum_{g' \in |\mathcal{G}|, g' \neq g} \frac{P(\tilde{z} = g|z = g')P(z = g')}{P(\tilde{z} = g)}$.

Now, $P(\tilde{z} = g) \geq P(\tilde{z} = g|z = g)P(z = g) \geq (1 - r)q$, and $P(\tilde{z} = g|z = g') \leq r$ for $g \neq g'$, and $\sum_{g' \neq g} P(z = g') \leq 1 - q$, so $P(z \neq \tilde{z}|\tilde{z} = g) \leq \frac{(1-q)r}{(1-r)q} \leq \frac{r}{(1-r)q}$. Also, of course $P(z \neq \tilde{z}|\tilde{z} = g) \leq 1$ as well, so $P(z \neq \tilde{z}|\tilde{z} = g) \leq \min\{\frac{r}{q(1-r)}, 1\}$.

$$
\begin{aligned}
&|\mathbb{E}[\ell(x, y)|\tilde{z} = g] - \mathbb{E}[\ell(x, y)|z = g]| = \\
&|\mathbb{E}[\ell(x, y)|\tilde{z} = g, z = g]P(z = g|\tilde{z} = g)+ \\
&\mathbb{E}[\ell(x, y)|\tilde{z} = g, z \neq g]P(z \neq g|\tilde{z} = g) - \mathbb{E}[\ell(x, y)|z = g]| \leq \\
&|\mathbb{E}[\ell(x, y)|\tilde{z} = g, z = g]P(z = g|\tilde{z} = g) - \mathbb{E}[\ell(x, y)|z = g]| + \\
&\mathbb{E}[\ell(x, y)|\tilde{z} = g, z \neq g]P(z \neq g|\tilde{z} = g) \leq \\
&|\mathbb{E}[\ell(x, y)|\tilde{z} = g, z = g]P(z = g|\tilde{z} = g) - \mathbb{E}[\ell(x, y)|z = g]| + B\min\{\tfrac{r}{q(1-r)}, 1\}.
\end{aligned}
$$

Similarly, $\mathbb{E}[\ell(x, y)|z = g] = \Big(\mathbb{E}[\ell(x, y)|\tilde{z} = g, z = g]P(\tilde{z} = g|z = g) + \mathbb{E}[\ell(x, y)|\tilde{z} \neq g, z = g]P(\tilde{z} \neq g|z = g)\Big)$, so

$$
\begin{aligned}
&|\mathbb{E}[\ell(x, y)|\tilde{z} = g, z = g]P(z = g|\tilde{z} = g) - \mathbb{E}[\ell(x, y)|z = g]| = \\
&\Big(|\mathbb{E}[\ell(x, y)|\tilde{z} = g, z = g]P(z = g|\tilde{z} = g) - \mathbb{E}[\ell(x, y)|\tilde{z} = g, z = g]P(\tilde{z} = g|z = g) - \\
&\mathbb{E}[\ell(x, y)|z \neq g, z = g]P(\tilde{z} \neq g|z = g)|\Big) \leq \\
&\mathbb{E}[\ell(x, y)|\tilde{z} = g, z = g]|P(z = g|\tilde{z} = g) - P(\tilde{z} = g|z = g)| + Br.
\end{aligned}
$$

Finally, we have $1 \geq P(z = g|\tilde{z} = g) = 1 - P(z \neq \tilde{z}|\tilde{z} = g) \geq \max\{1 - \frac{r}{q(1-r)}, 0\}$, and $1 \geq P(\tilde{z} = g|z = g) \geq 1 - r$, so

$|P(z=g|\tilde{z}=g) - P(\tilde{z}=g|z=g)| \leq \min\{\frac{r}{q(1-r)}, 1\}$. Thus, altogether we have $|\mathbb{E}[\ell(x,y)|\tilde{z}=g] - \mathbb{E}[\ell(x,y)|z=g]| \leq B\min\{\frac{r}{q(1-r)},1\} + B\min\{\frac{r}{q(1-r)},1\} + Br \leq 3B\min\{\frac{r}{q(1-r)},1\}$, which is $O(r/q)$ in terms of r and q.

C.5.2. PER-ESTIMATED-GROUP POPULATION LOSS VS. TRAINING LOSS

Let $\tilde{S}_g$ denote the set of training points with predicted group label g. The datapoints in $\tilde{S}_g$ are independent samples from $P(x,y|\tilde{z}=g)$. So by Theorem C.5, for a given g we have that the difference between the per-group training and population losses on the Stage 2 classification task is $|\mathbb{E}_{(x,y)\sim\tilde{S}_g}[\ell(x,y;\theta)] - \mathbb{E}[\ell(x,y;\theta)|\tilde{z}=g]| \leq O\left(\sqrt{\frac{\log|\tilde{S}_g|}{|\tilde{S}_g|}}\right)$ with high probability.

By Hoeffding's inequality we have $O\left(\sqrt{\frac{\log|\tilde{S}_g|}{|\tilde{S}_g|}}\right) \leq O\left(\sqrt{\frac{\log n}{q(1-r)n}}\right) = O\left(\sqrt{\frac{\log n}{qn}}\right)$. Thus by triangle inequality and union bound, we have $|\mathbb{E}_{(x,y)\sim\tilde{S}_g}[\ell(x,y;\theta)] - \mathbb{E}[\ell(x,y;\theta)|z=g]| \leq O(r/q) + O\left(\sqrt{\frac{\log n}{qn}}\right) = \tilde{O}\left(\frac{r}{q} + \frac{1}{\sqrt{qn}}\right)$ for all $g \in \mathcal{G}$ and all θ with $\|\theta\|_2 \leq R$ with high probability. Finally, applying Lemma C.6 yields the desired result, as the training loss is $\max_{g\in\mathcal{G}} \mathbb{E}_{(x,y)\sim\tilde{S}_g}[\ell(x,y;\theta)]$. □

C.6. Extension of Theorem 4.1: Non-realizable case

In this section, we show that if the group classifier makes randomized predictions according to the predicted probabilities (rather than classifying groups by picking the group with maximum predicted probability), we can yield a more general bound than Theorem 4.1 that is better in the non-realizable case (i.e., when there does not exist a perfect group classifier). We relax the realizability requirement stated in Section 4 to the requirement that there must exist some $f^* \in \mathcal{F}_1$ such that $f(x,y)$ is the vector of true probabilities $P(z|x,y)$. This theorem is stated below.

Theorem C.8. *Suppose there exists $f^* \in \mathcal{F}_1$ such that $f(x,y)$ is the vector of true probabilities $P(z|x,y)$. Let $\hat{P}(z|x,y)$ denote the probabilities output by the group classifier from "Stage 1," and suppose for each datapoint (x,y) the predicted group label $\tilde{z}$ is sampled from $\hat{P}(z|x,y)$. Suppose that the total variation between $\hat{P}(z|x,y)$ and $P(z|x,y)$ is bounded by r, for all x,y in the support of P. Then with high probability, $\mathcal{L}_{robust}(\hat{\theta}_{robust}) \leq \mathcal{L}^*_{robust} + \tilde{O}\left(\frac{r}{q} + \frac{1}{\sqrt{qn}}\right)$.*

Theorem C.8 shows that even if it is impossible to perfectly distinguish the groups, this is not necessarily an obstacle to the downstream robust performance. To prove Theorem C.8, we first prove the following lemma.

Lemma C.9. *Suppose $P(\tilde{z}|x,y) = P(z|x,y)$. Then $\mathbb{E}[\ell(x,y;\theta)|z=g] = \mathbb{E}[\ell(x,y;\theta)|\tilde{z}=g]$, for all $g \in \mathcal{G}$.*

Proof. By Bayes' rule $P(x,y|\tilde{z}=g) = P(\tilde{z}=g|x,y)P(x,y)/P(\tilde{z}=g)$. By assumption, for any (x,y), $P(\tilde{z}=g|x,y) = P(z=g|x,y)$. Therefore $P(x,y|\tilde{z}=g) = P(\tilde{z}=g|x,y)P(x,y)/P(\tilde{z}=g) = P(z=g|x,y)P(x,y)/P(z=g) = P(x,y|z=g)$ by applying Bayes' rule again. The claim follows. □

Lemma C.9 implies that if we use the predicted group labels $\tilde{z}$ rather than the "true" group labels z, there is essentially no difference since $\tilde{z}$ and z have the same distribution conditioned on x,y. Thus, samples from $(x,y,\tilde{z})$ and (x,y,z) are equivalent for our purposes, and applying Lemma C.1 shows that the minimizer of $\hat{\tilde{\theta}}$ of $\max_g \mathbb{E}[\ell(x,y;\theta)|\tilde{z}=g]$ satisfies $\mathcal{L}_{robust}(\hat{\tilde{\theta}}) \leq \mathcal{L}^*_{robust} + \tilde{O}(\frac{1}{\sqrt{qn}})$ with high probability (since in this case we have n total datapoints).

Proof of Theorem C.8. Note that if $r=1$ the statement follows trivially from boundedness. Similarly if $r=0$ the statement follows from the argument above. So assume $0<r<1$.

Given x,y, suppose we sample $z', \tilde{z}'$ in the following "coupled" fashion. Flip a biased coin with probability of heads being $\sum_{g\in\mathcal{G}} \min\left\{\hat{P}(z=g|x,y), P(\tilde{z}=g|x,y)\right\}$.

If the coin is heads, then sample z'' from the distribution where

$$P(z''=g) = \frac{\min\{\hat{P}(z=g|x,y), P(\tilde{z}=g|x,y)\}}{\sum_{g\in\mathcal{G}} \min\{\hat{P}(z=g|x,y), P(\tilde{z}=g|x,y)\}}.$$

and set $z' = \tilde{z}' = z''$. Note that by the assumption $r < 1$, the denominator is nonzero.

If the coin is tails, then sample z' from the distribution where

$$P(z'=g) = \frac{P(z=g|x,y) - \min\{\hat{P}(z=g|x,y), P(z=g|x,y)\}}{1 - \sum_{g\in\mathcal{G}} \min\{\hat{P}(z=g|x,y), P(\tilde{z}=g|x,y)\}},$$

and independently sample $\tilde{z}'$ from the distribution where

$$P(\tilde{z}'=g) = \frac{P(\tilde{z}=g|x,y) - \min\{\hat{P}(z=g|x,y), P(\tilde{z}=g|x,y)\}}{1 - \sum_{g\in\mathcal{G}} \min\{\hat{P}(z=g|x,y), P(\tilde{z}=g|x,y)\}}.$$

Also note that by the assumption $r > 0$, the denominator is nonzero.

Notice that using this sampling procedure, $P(\tilde{z}'|x,y) = P(z|x,y)$. Similarly, $P(z'|x,y) = \hat{P}(z'|x,y)$. Thus by Lemma C.9, $\mathbb{E}[\ell(x,y;\theta)|z'=g] = \mathbb{E}[\ell(x,y;\theta)|z=g] = \mathcal{L}_{robust}(\theta)$, and $\mathbb{E}[\ell(x,y;\theta)|\tilde{z}'=g] = \mathbb{E}[\ell(x,y;\theta)|\tilde{z}=g]$. Also observe that the probability that $z' \neq \tilde{z}'$ is $\leq 1 - \sum_{g\in\mathcal{G}} \min\{\hat{P}(z=g|x,y), P(\tilde{z}=g|x,y)\}$ which is precisely the total variation between $\hat{P}(z=g|x,y)$ and $P(\tilde{z}=g|x,y)$. From here, the remainder of the proof is essentially identical to the proof of Theorem 4.1, as the sampled group labels z' from the true conditional distribution are "equivalent" to "correct" group labels, and the disagreement rate between z' and the estimated group labels $\tilde{z}'$ is $\leq r$ by the total variation assumption.

C.7. Proof of Corollary 4.2

Corollary C.10. *With high probability, for* BARACK*-Base we have* $\mathcal{L}_{robust}(\hat{\theta}_{robust}) \leq \mathcal{L}^*_{robust} + \tilde{O}\left(\frac{1}{q\sqrt{m}}\right)$.

Proof. First, by Lemma C.1, we have that if we train a classifier with group DRO on $\mathcal{D}_1$ to classify the *group labels*, the worst-group population loss is $\tilde{O}(1/\sqrt{qm})$ with high probability (since in the realizable case, there exists a group classifier with 0 population loss). As discussed in Appendix C.4.1, for both the cross-entropy and the squared loss this translates to a (population) worst-group misclassification *error* of $\tilde{O}(1/\sqrt{qm})$ as well (when the classifier prediction is the group with maximum predicted probability).

In fact, by inspecting the proof of Lemma C.1, we can make the slightly stronger statement that the population classification error on group g is $\tilde{O}(1/\sqrt{q_g m})$ with high probability.

So, $P(z \neq g|\tilde{z}=g) = \sum_{g'\in|\mathcal{G}|, g'\neq g} \frac{P(\tilde{z}=g|z=g')P(z=g')}{P(\tilde{z}=g)} =$

$\sum_{g'\in|\mathcal{G}|, g'\neq g} \frac{\tilde{O}(1/\sqrt{q_{g'}m})\cdot q_{g'}}{P(\tilde{z}=g)} \leq O\left(\frac{1}{(1-\tilde{O}(1/\sqrt{q_g m}))q_g\sqrt{m}} \sum \sqrt{q_{g'}}\right) = \tilde{O}(\frac{1}{q\sqrt{m}})$ since $P(\tilde{z}=g) \geq P(\tilde{z}=g|z=g)P(z=g) \geq (1-\tilde{O}(1/\sqrt{q_g m}))q_g$ with high probability and $\sum\sqrt{q'_g} \leq \sum 1/\sqrt{|\mathcal{G}|} = \sqrt{|\mathcal{G}|}$. Thus by a similar argument to the proof of Theorem 4.1, we have $|\mathbb{E}[\ell(x,y)|\tilde{z}=g] - \mathbb{E}[\ell(x,y)|z=g]| \leq \tilde{O}(\frac{1}{q\sqrt{m}})$, and in turn can obtain the desired result (by an analogous argument as Appendix C.5.2).

C.8. Proof of Corollary C.4

Corollary C.4 is restated below.

Corollary C.11. *In addition to the assumptions of Theorem 4.1, suppose that for all* $f \in \mathcal{F}_2$, $\ell(f(x),y) \perp \text{argmax}\{\hat{f}_{group}(x,y)\} \mid z$. *Let* θ^*_{avg} *be the minimizer of the population* average *loss* $\mathbb{E}_{(x,y)\sim P}[\ell(x,y;\theta)]$. *Then for* BARACK*-Base we have* $\mathcal{L}_{robust}(\hat{\theta}_{robust}) \leq \min\left(\mathcal{L}^*_{robust} + \tilde{O}(\frac{1}{q\sqrt{m}}), \mathcal{L}_{robust}(\theta^*_{avg}) + \tilde{O}(\frac{1}{\sqrt{qn}})\right)$ *w.h.p.*

Proof. In words, the assumption that $\ell(f(x), y) \perp \text{argmax}\{\hat{f}_{group}(x, y)\} \,\big|\, z$ for all $f \in \mathcal{F}_2$ says that for all classifiers in $\mathcal{F}_2$, the loss on the *target task* (of classifying the class labels) is independent of the prediction of the group classifier, when conditioned on the actual group label.

Using this assumption, we have that $\mathbb{E}_{(x,y)}[\ell(x, y; \theta)|\tilde{z} = g] =$
$\mathbb{E}_{(x,y)}[\ell(x, y; \theta)|\tilde{z} = g, z = g]P(z = g|\tilde{z} = g) + \mathbb{E}_{(x,y)}[\ell(x, y; \theta)|\tilde{z} = g, z \neq g] \cdot P(z \neq g|\tilde{z} = g) =$
$\mathbb{E}_{(x,y)}[\ell(x, y; \theta)|z = g]P(z = g|\tilde{z} = g) + \mathbb{E}_{(x,y)}[\ell(x, y; \theta)|z \neq g]P(z \neq g|\tilde{z} = g)$.

Denote $\mathcal{L}_{avg}(\theta) = \mathbb{E}_{(x,y)}[\ell(x, y; \theta)]$. Note that $\mathbb{E}_{(x,y)}[\ell(x, y; \theta)|z \neq g]P(z \neq g) + \mathbb{E}_{(x,y)}[\ell(x, y; \theta)|z = g]P(z = g) = L_{avg}(\theta)$. Thus,

$$\mathbb{E}_{(x,y)}[\ell(x, y; \theta)|z \neq g]P(z \neq g|\tilde{z} = g) =$$
$$\frac{P(z \neq g|\tilde{z} = g)}{P(z \neq g)} \left(L_{avg}(\theta) - \mathbb{E}_{(x,y)}[\ell(x, y; \theta)|z = g]P(z = g)\right).$$

As a result,

$$\mathbb{E}_{(x,y)}[\ell(x, y; \theta)|\tilde{z} = g] = (1 - a)\mathbb{E}_{(x,y)}[\ell(x, y; \theta)|z = g] + aL_{avg}(\theta),$$

where $a = \dfrac{P(z \neq g|\tilde{z} = g)}{P(z \neq g)} \leq \tilde{O}(\frac{1}{\sqrt{qm}})$ with high probability (as shown in previous sections).

Let $\tilde{\theta}_{pop}$ denote the minimizer of $\mathbb{E}_{(x,y)}[\ell(x, y; \theta)|\tilde{z} = g]$. Suppose for contradiction that $\mathcal{L}_{robust}(\tilde{\theta}_{pop}) > \mathcal{L}_{robust}(\theta^*_{avg})$. Let

$$k_{max} \in \underset{k \in \mathcal{G}}{\text{argmax}} \left((1 - a) \cdot \mathbb{E}_{(x,y)\sim p_k}[\ell(x, y; \theta)] + a \cdot \mathbb{E}_{(x,y)\sim P}[\ell(x, y; \theta)]\right).$$

Then we have $\left((1 - a) \cdot \mathbb{E}_{(x,y)\sim p_{k_{max}}}[\ell(x, y; \theta)] + a \cdot \mathbb{E}_{(x,y)\sim P}[\ell(x, y; \theta)]\right) > \mathbb{E}_{(x,y)\sim p_{k_{max}}}[\ell(x, y; \theta)]$, i.e. that $\mathbb{E}_{(x,y)\sim P}[\ell(x, y; \theta)] > \mathbb{E}_{(x,y)\sim p_{k_{max}}}[\ell(x, y; \theta)]$ and thus that $\left((1 - a) \cdot \mathbb{E}_{(x,y)\sim p_{k_{max}}}[\ell(x, y; \theta)] + a \cdot \mathbb{E}_{(x,y)\sim P}[\ell(x, y; \theta)]\right) < \mathbb{E}_{(x,y)\sim P}[\ell(x, y; \theta)]$. But there must be some $k \in [\mathcal{G}]$ such that $\left((1 - a) \cdot \mathbb{E}_{(x,y)\sim p_{k_{max}}}[\ell(x, y; \theta)] + a \cdot \mathbb{E}_{(x,y)\sim P}[\ell(x, y; \theta)]\right) \geq \mathbb{E}_{(x,y)\sim P}[\ell(x, y; \theta)]$, which contradicts the definition of k_{max}. Thus, $\mathcal{L}_{robust}(\tilde{\theta}_{pop}) \leq \mathcal{L}_{robust}(\theta^*_{avg})$.

Using the fact that the datapoints in $\tilde{S}_g$ are independent samples from $P(x, y|\tilde{z} = g)$, we have for all g that $|\mathbb{E}_{(x,y)\sim\tilde{S}_g}[\ell(x, y; \theta)] - \mathbb{E}[\ell(x, y; \theta)|\tilde{z} = g]| \leq O\left(\sqrt{\frac{\log|\tilde{S}_g|}{|\tilde{S}_g|}}\right)$ with high probability (as argued in Appendix C.5.2). By combining this with the result of Corollary 4.2, we obtain the desired result. □

C.9. Proof of Lemma C.2

Lemma C.12. *Let $\hat{f}_{erm,full} \in \mathcal{F}_2$ (with associated parameters $\hat{\theta}_{erm,full}$) denote the ERM classifier trained on the full dataset. There exists a distribution $\mathcal{P}$ such that with high probability, $\mathcal{L}_{robust}(\hat{\theta}_{erm,full}) \geq \frac{\log(1/q)}{\log(|\mathcal{Y}|)}\mathcal{L}^*_{robust} - \tilde{O}(1/\sqrt{n})$.*

Proof. Consider the following simple distribution. Suppose that the distribution of x is a point mass on a single point. Suppose that there are k classes and the classes are identical to the groups (that is, $y = z$ always). If cross-entropy loss is used, then clearly to minimize worst-group loss one should predict a uniform distribution over each class (since the point x gives no information about the class or group). Thus, the loss on each point would be $\log k$, so $\mathcal{L}^*_{robust} = \log k$ in this case.

Recall that the cross-entropy loss has the property that the *average* population cross-entropy loss is minimized when the predicted probability of each class is simply the true probability of that class conditioned on the features x. In our case, the latter is simply the probability of the class. Thus, for the minimizer of the population average cross entropy loss, the predicted class probabilities for any point are the true class probabilities $P(c)$, and so the loss given that the true class is c is $-\log P(c)$ by definition. So the worst-group loss is $\max_c -\log P(c) = -\log \min_c P(c) = \log(1/q)$ [note that $1/q \geq k$]. The lemma now follows by a simple application of Hoeffding's inequality, since the training dataset has n points.

C.10. Theoretical Analysis of BARACK-SSL

In Corollary C.13 (below), we analyze BARACK-SSL (Algorithm 5), which uses semi-supervised learning to train the group classifier and is described further in Appendix B. For this corollary, we assume that the group classification problem is realizable (i.e., there exists a classifier with 0 training loss on the group classification task) and that $m = \Omega(\sqrt{n})$.

Corollary C.13. *Under appropriate conditions, for* BARACK-*SSL we have* $\mathcal{L}_{robust}(\hat{\theta}_{robust}) \leq \mathcal{L}^*_{robust} + \tilde{O}\left(\frac{1}{q\sqrt{n}} + \frac{1}{qm}\right)$ *with high probability.*

In BARACK-SSL, *semi-supervised* learning is used to train the group classifier, leveraging the group-unlabeled points. With an appropriate choice of semi-supervised learning method (such as FixMatch (Sohn et al., 2020)), if $m = \Omega(\sqrt{n})$, then under appropriate conditions, the worst-group generalization error rate bound of BARACK-SSL is $O(1/\sqrt{n})$. This result is based on the PAC-learning based results of (Balcan & Blum, 2009), and requires the assumptions therein as well as realizability (the exact conditions are somewhat technical, and are deferred to the discussion in Appendix C). This translates to an excess robust risk of $O(1/\sqrt{n})$ for the final model, as stated in Corollary C.13.

To prove Corollary C.13, we make use of the following theorem (see Theorem C.14 below, a restated version of Theorem 21.8 from (Balcan & Blum, 2009)). First, we need some additional notation: let $\chi : \mathcal{F}_1 \times \mathcal{X} \to \{0, 1\}$ be a function and define the overloaded notation $\chi(f, P) = \mathbb{E}_{x\sim P}[\chi(f, x)]$. Let $VC(\mathcal{C})$ denote the VC-dimension of a function class $\mathcal{C}$. For function classes that output probabilities rather than labels directly, we overload notation so that $VC(\mathcal{F}) = VC(\{\text{argmax}\{f\} \mid f \in \mathcal{F}\})$. Recall that $\mathcal{D}_2$ denotes the training set of group-unlabeled points.

Theorem C.14. *Given* $\delta \in (0, 1)$*, if* $n = \Omega\left(\frac{1}{\epsilon^2}\left(VC(\chi(\mathcal{F}_1)) \log \frac{1}{\epsilon} + \log \frac{1}{\delta}\right)\right)$ *and* $m \geq \frac{2}{\epsilon}\left(\log(2s(2m, t + 2\epsilon)) + \log \frac{2}{\delta}\right)$*, then with probability* $\geq 1 - \delta$ *it holds that all* $f \in \mathcal{F}_1$ *with zero training error and* $1 - \chi(f, \mathcal{D}_2) \leq t + \epsilon$ *have population error* $\leq \epsilon$*. Here,* $s(2m, t + 2\epsilon)$ *denotes the expected number of splits when* $2m$ *points are drawn IID from* $\mathcal{P}$ *with concepts* $f \in \mathcal{F}_1$ *having* $\chi(f, \mathcal{P}) \leq t + 2\epsilon$*.*

FixMatch (Sohn et al., 2020) (which we use for Stage 1 of BARACK-SSL, as described in 5) minimizes a weighted sum of the *supervised* loss (computed on the training points with group labels) and an *unsupervised* loss (consistency of predictions between examples and augmented versions of the same example, computed on all training points). Concretely, in our case the FixMatch loss for a group classifier $f \in \mathcal{F}_1$ is $\mathcal{L}_{fixmatch} = \lambda\mathcal{L}_{sup} + (1 - \lambda)\mathcal{L}_{consistency}$, where $\lambda \in [0, 1]$, $\mathcal{L}_{sup} = \mathbb{E}_{(x,y,z)\sim\mathcal{D}_1}[\ell(f(x, y), z)]$, and $\mathcal{L}_{consistency} = \mathbb{E}_{(x,y)\sim\mathcal{D}_2}[\mathbf{1}(\max\{f(x, y)\} \geq \tau) \cdot \ell(f(aug(x), y), f(x, y))]$, where $\tau \in [0, 1]$ is a predefined constant.

Here, $aug(\cdot)$ denotes the augmentation function; for simplicity assume it is a fixed (non-random) function. Let $\chi(f, x)$ be 0 if $\max\{f(x)\} \geq \tau$ and $\text{argmax}\{f(x)\}$ $\neq \text{argmax}\{f(aug(x))\}$, and 1 otherwise (in other words, $\chi(f, x)$ is 1 unless f makes a confident prediction on x but makes a different prediction on the augmented version of x). By the realizability assumption, there exists $f \in \mathcal{F}_1$ with zero loss (and therefore zero training loss) on the supervised task of classifying the groups. Note that if aug is the identity function, then f also attains zero consistency loss; more generally, given $\xi \geq 0$ we can choose aug to be a "weak enough" augmentation such that there exists $f \in \mathcal{F}_1$ such that $\mathcal{L}_{consistency} \leq \xi$. If we choose ξ to be sufficiently small, we can guarantee that $\mathcal{L}_{consistency} \leq \delta$ implies $\chi(f, \mathcal{D}_2) \leq t + \epsilon$, since $t, \epsilon \geq 0$.

Because there exists $f \in \mathcal{F}_1$ with zero loss, this implies that if we set the weight λ on the supervised part of the loss to be large enough, the function $\mathcal{F}_1$ we learn (corresponding to the minimizer of $\mathcal{L}_{fixmatch}$) will have zero training error. (If we find a function with nonzero training error, we can increase the weight of the supervised loss and rerun.) Then, invoking Theorem C.14, we get that the *population* error of the returned group classification model will be $\leq O(\epsilon)$ with high probability as long as m, n satisfy $m = \Omega(1/\epsilon^2)$ and $n = \Omega(1/\epsilon)$. The value of the term t in Theorem C.14 depends on the consistency error $\chi(f, \mathcal{D}_2)$ on the training data, which in turn will depend on the augmentation chosen, the threshold τ, and the weight $1 - \lambda$ (smaller λ will encourage lower consistency loss). Similarly, the constants in the preceding $\Omega(\cdot)$ terms depend on the choice of augmentation function $aug(\cdot)$; intuitively, we would like to choose an augmentation function that is as strong as possible while still being label-preserving (so as not to make the consistency loss large).

In summary, given the required conditions, if $m = \Omega(1/\epsilon^2)$ and $n = \Omega(1/\epsilon)$ then the population error of the returned group classification model is $\leq O(\epsilon)$ with high probability. The population error of the group classifier on each group g will then be $\leq O(\epsilon/q_g)$ with high probability. Equivalently, with high probability, for all groups g the population error of the group classifier on that group will be $O(\frac{1}{q_g m} + \frac{1}{q_g\sqrt{n}})$ [as long as $m = \Omega(\sqrt{n})$]. Finally, the desired result now follows by invoking Theorem 4.1.

D. Experimental Details and Additional Results

We first describe the datasets, baselines, and experimental details more thoroughly. We then present extensive ablation experiments in Appendix D.3 and results for BARACK-SSL in Appendix D.4 to support the robustness and extensibility of our BARACK framework.

D.1. Dataset Details

D.1.1. DATASETS

Waterbirds. Waterbirds (Sagawa et al., 2019), is a popular robustness benchmark that consists of images from 'landbird' and 'waterbird' species on either land or water backgrounds. The task is to classify images as 'landbird' vs. 'waterbird', and the groups are defined by background.[5] 95% of landbirds are on land backgrounds and similarly for waterbirds; this spurious correlation makes landbirds on water and waterbirds on land harder to classify.

CelebA. CelebA (Liu et al., 2015a) is a popular face classification dataset which is also often used to evaluate robustness to spurious correlations. The task is to classify faces as 'blond' or 'not blond', and the groups are defined by gender. Blondness is spuriously correlated with gender: the female faces are disproportionately blond and the male faces disproportionately non-blond. For instance, only 6% of blond examples are male, leading to poor performance on this group.

U-MNIST. U-MNIST (Sohoni et al., 2020) is a modified version of MNIST (LeCun et al., 2010), where the task is to classify digits as '< 5' or '≥ 5', the groups are the individual digits, and only 5% of images in the '8' group are retained from the individual dataset. This rarity makes '8' images more difficult to classify.

U-CIFAR10. We introduce U-CIFAR10 as a modification of the CIFAR-10 dataset (Krizhevsky, 2009), where the task is to classify the image as 'animal' or 'vehicle', the groups are the 10 original CIFAR-10 classes, and we undersample the 'airplane' class to 5%. Though similar to U-MNIST, this task is much more challenging.

D.1.2. DATASET SETUP

Recall that for our experiments, we pick a fixed budget of (training and validation) examples to label for each group, which are randomly sampled from the appropriate group in the original dataset, and evaluate how our approach fares as this budget is increased. We use the same group label budget for both the training and validation sets. Note that it is not fundamentally necessary for the group-labeled training and validation subsets to be the same size; this is merely a simple heuristic to trade off the amount of data for training the group classifier, and the amount of data for model selection.

Note that in reality, a group-labeled dataset may not have the same number of labeled points for each group, but rather reflect the training/population distribution, i.e. have the same group proportions as the population. In this case, however, we can still convert our group-labeled dataset to one with balanced numbers of points per group, by simply subsampling the group-labeled points by group to get a (smaller) balanced dataset; the performance we achieve after this subsampling can be interpreted as a rough lower bound on the performance without such subsampling, since it simply throws away data. In fact, we empirically observed that this subsampling does not meaningfully degrade the final performance, compared to using the un-subsampled version (containing more group-labeled points for the larger groups).

[5] Concretely, for our purposes we have four groups: landbird on land, landbird on water, waterbird on land, and waterbird on water, as in (Sagawa et al., 2019).

D.2. Training Details

For all datasets and methods, we use a fixed training/validation/test split. Models are trained on the training set, and the validation set is used for model selection (both selecting the best model during training, and for hyperparameter selection). All results reported in plots and tables are on the test set; the test set is **not** used for any model selection or tuning purposes. For consistency and direct comparability with prior works, we do not use data augmentation for any of the baselines or while training the "Stage 2" robust models, except when explicitly specified otherwise. For all experimental settings, reported means and standard deviations are over 5 trials with different random seeds. (We also average over 5 trials for all ablation experiments in Appendix D.3.)

U-MNIST. This task is based on the MNIST dataset (LeCun et al., 2010) (available under the Creative Commons Attribution-Share Alike 3.0 license). The U-MNIST task is to classify digits between ≤ 4 and ≥ 5; the groups are the individual digits. We use a fixed training-validation split for all methods, to set aside 20% of the original MNIST training set (12,000 points) for validation. On the training set, the '8' digits are subsampled such that only 5% of them are kept. The total number of points with each digit label in the training set are $0:4769, 1:5382, 2:4845, 3:4950, 4:4662, 5:4300, 6:4728, 7:4980, 8:234, 9:4720$. The validation set is approximately balanced. Without modification, this can actually help methods that do not require group labels on the full validation set (such as ERM, BARACK, GEORGE) to more easily select models with good worst-group performance, because the rare '8' group is overrepresented in the validation set compared to the training set, so it is easier to detect poor performance on that group even when looking at overall performance or performance using noisy group labels. Thus, when computing an average metric (loss or accuracy) on a subset of the validation data, we compute a weighted average which is the sum over each group g of: the average of that metric for all points in the subset whose true group label is g, times the proportion of group g in the training dataset. This reweighting procedure is the same as what is done in (Sagawa et al., 2019; Sohoni et al., 2020).

We use a 4 layer LeNet (LeCun et al., 1998) and the Adam optimizer for all methods. For U-MNIST, for training the robust model we train for 100 epochs with a batch size of 128, the Adam optimizer (Kingma & Ba, 2015), and decay the learning rate by a factor of 0.1 at epochs 50 and 75. These hyperparameters were taken from (Sohoni et al., 2020). We tune all methods over the cross product of learning rates [2e-3, 2e-4] and weight decays [1e-4, 3e-4, 1e-5]. For methods using GDRO for the second stage, we also tune the GDRO group adjustment parameter in the set $\{0, 3\}$, and use uniform per-group sampling, as described in (Sagawa et al., 2019).

U-CIFAR10. This task is based on the publicly available CIFAR-10 dataset (Krizhevsky, 2009) (license unknown). The U-CIFAR10 task is to classify images as "vehicle" or "animal"; the groups are the original CIFAR-10 classes. We use a fixed training-validation split for all methods, to set aside 20% of the original MNIST training set (10,000 points) for validation. On the training set, the 'airplane' images are subsampled such that only 5% of them are kept. The total number of points with each group label in the training set are airplane: 204, automobile: 4004, bird: 3976, cat: 4017, deer: 3997, dog: 3999, frog: 4000, horse: 3976, ship: 3957, truck: 4003. The validation set is approximately balanced, so as described for U-MNIST we compute reweighted metrics where appropriate.

We use a ResNet-50 (He et al., 2016) model and train for 200 epochs with a batch size of 128, SGD with momentum 0.9, and a cosine learning rate schedule. These hyperparameters were taken from the implementation at `https://github.com/kuangliu/pytorch-cifar`. We tune all methods over the cross product of learning rates [1e-2, 1e-3] and weight decays [1e-3, 3e-3, 1e-2, 3e-2, 1e-1]. For methods using GDRO for the second stage, we also tune the GDRO group adjustment parameter in the set $\{0, 3\}$, and use uniform per-group sampling, as described in (Sagawa et al., 2019).

Waterbirds. The Waterbirds dataset was created by (Sagawa et al., 2019) as a modification of the CUB dataset (Wah et al., 2011) (license unknown). It consists of different bird species (with class labels either "waterbird" or "landbird") on either a land or water background. There are 3498 training images of landbirds on land, 184 of landbirds on water, 56 of waterbirds on land, and 1057 of landbirds on land (these are the four groups). The validation set is more balanced, so as described for U-MNIST we compute reweighted metrics where appropriate.

We use a ResNet-50 (He et al., 2016) model and train for 300 epochs with a batch size of 128 and SGD with momentum 0.9. These hyperparameters were taken from (Sagawa et al., 2019). We tune all methods over the (learning rate, weight decay) pairs (1e-4, 1e-1), (1e-3, 1e-4), and (1e-5, 1.0), as done in (Sagawa et al., 2019; Liu et al., 2021). For methods using GDRO for the second stage, we set the GDRO group adjustment parameter to 2 as in (Sagawa et al., 2019), and use uniform per-group sampling.

CelebA. The CelebA dataset (Liu et al., 2015b) (license unknown) is a dataset of celebrity faces annotated with several descriptors (such as gender, hair color, wearing glasses). It is often used as a benchmark for robustness to spurious correlations. The task we consider is classifying the hair color of the person in the image as blond or non-blond, as in prior works such as (Sagawa et al., 2019; Sohoni et al., 2020; Liu et al., 2021). In this dataset, hair color is spuriously correlated with gender: there are 71629 images in the "female, non-blond" group, 66874 in the "female, blond", 22880 "male, non-blond", and just 1387 "male, blond".

We use a ResNet-50 (He et al., 2016) model and train for 50 epochs with a batch size of 128 and SGD with momentum 0.9. These hyperparameters were taken from (Sagawa et al., 2019). We tune all methods over the (learning rate, weight decay) pairs (1e-4, 1e-2), (1e-4, 1e-4), and (1e-5, 0.1), as done in (Sagawa et al., 2019; Liu et al., 2021). For methods using GDRO for the second stage, we set the GDRO group adjustment parameter to 3 as in (Sagawa et al., 2019), and use uniform per-group sampling.

D.2.1. BASELINE DETAILS

We reimplemented ERM and GDRO (and subset-GDRO) ourselves, along with BARACK. For the other baseline methods (George (Sohoni et al., 2020), JTT (Liu et al., 2021), and EIIL (Creager et al., 2021)), we use the authors' publicly available repositories, adapting the code (such as to plug in our dataloaders) where necessary.

All three methods require first training an ERM model. For GEORGE, this model is either a standard ERM model or one trained with high regularization, and is selected based on the Silhouette score of the clustered activations. For U-CIFAR10 (the only dataset which George did not evaluate on originally), we used a learning rate of 1e-3 for this model and tuned the weight decay in [1e-3, 1e-1] based on this Silhouette score criterion. For JTT and EIIL, the model is an ERM model trained with high regularization for a shorter number of epochs. For U-MNIST and U-CIFAR10 (which the JTT paper did not evaluate on originally) we used a learning rate of 2e-3 and tuned the weight decay in [1e-1, 1e-3] for this ERM model, and tuned the number of training epochs in $\{1, 50\}$ (the tuning criterion in this case was the validation worst-group accuracy of the final robust model), for both EIIL and JTT. EIIL also did not evaluate on CelebA, so we use the same ERM model as in JTT.

For JTT and EIIL, we assume the group labels on the entire validation set are known, and use these for model selection (as done in the respective papers). This gives them a slight advantage for model selection compared to BARACK, in which we only use a small number of group-labeled validation examples (the same number as we use for training). On the other hand, George does not assume validation set group labels, but rather estimates them the same way the training group labels are estiated (via clustering the activations of the last layer).

D.2.2. BARACK DETAILS

For BARACK-base, we train a supervised group classifier model (with GDRO) to predict group pseudolabels, as described in Section 3. This is challenging due to the low number of points with known group labels: for instance, with 8 group-labeled points per group on Waterbirds or CelebA, the entire training dataset for this stage is 32 examples. Thus, we generally need to train for more epochs to obtain reasonable results. We also use mild data augmentation (random crops and flips) for the Stage 1 group classifier only to help deal with this lack of data (although, similarly to prior works, we do not use data augmentation in Stage 2, except for the experiment with RotNet with data augmentation).

For the learning rate and weight decay for the group classifier, we tune over the same pairs of hyperparameters as described above. For the number of epochs, we train for $500/(n_{lab}/64)$ epochs, where n_{lab} is the number of group-labeled examples in the smallest group (in our experiments, we evaluate the settings $n_{lab} \in \{8, 16, 32, 64\}$). (We evaluate the validation accuracy every $n_{lab}/64$ epochs to keep a constant number of validation evaluations.) We set the batch size for training the group classifier to be the minimum of 128 or the total number of group-labeled examples.

We note that the group classifier and end model need not have the same architecture. For instance, one could potentially select a smaller model for the group classifier since it is trained with a small amount of data. In our preliminary experiments, however, we found that using the larger ResNet-50 model (the same architecture as the Stage 2 model) performed better than using a smaller ResNet-18 (by 5-10% group classification accuracy on average across all datasets).

For BARACK (and Subset-GDRO), the points for which we know the group label are selected randomly from the training and validation datasets. We select different sets of these points depending on the random seed (but for a fixed seed, these sets are the same, to facilitate direct comparisons). We do this to avoid over-indexing interpretation of results to a particularly

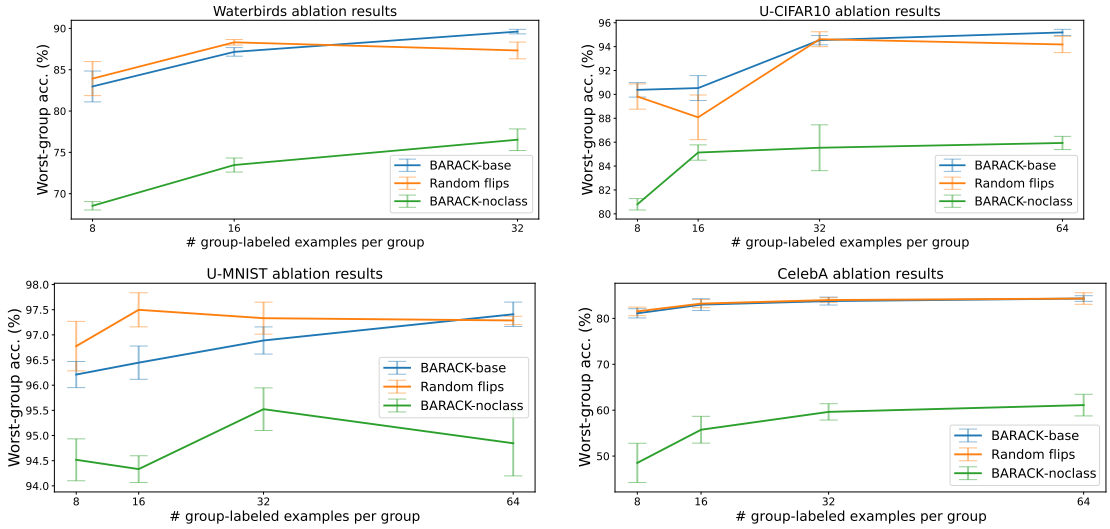

Figure 8. Worst-group accuracy for random flipping experiment (orange line) and no-class experiment (green line).

"easy" or "difficult" set chosen by happenstance. In the real world, of course, the points for which the group labels are known would generally be a fixed set. We note that hyperparameters should not be tuned by looking at averaged metrics over the different seed results, because in a way this "shares information" between different trials with different group-labeled points, so in some sense is using more group label information than it should. Thus, we indeed select the best hyperparameters separately for each seed (based on the appropriate validation metric for that seed). For consistency, we do this for *all* methods (e.g., tune hyperparameters on a per-seed basis based on the validation metric).

D.2.3. INITIALIZATION

For training both the group classifier and the robust model, we typically start from a pretrained model: ResNet-50 pretrained on ImageNet. The exception is that for U-MNIST, we use a LeNet-5. In Appendix D.3.3, we evaluate the impact of using different pretrained models (for the tasks other than U-MNIST).

D.3. Ablation Experiments

In this section, we present ablation experiments to study the reasons behind the worst-group accuracy gains offered by BARACK. First, we run a synthetic experiment in which we run GDRO with randomly generated noisy group labels at different noise levels, to better understand how group prediction errors affect the final robust performance. Next, we ablate the importance of using the class label as an input to the group prediction model (as described in Section 3). Finally, we explore using models that are pretrained on ImageNet in a *self-supervised* manner (instead of supervised) as the starting model for BARACK (and all the baselines), to understand how the worst-group accuracy trends from Section 5.2 generalize when different pretrained models are used.

D.3.1. RANDOM FLIPPING.

To better understand why BARACK can achieve worst-group accuracy close to that of GDRO even with fairly inaccurate predicted group labels, we run a synthetic experiment: we run GDRO with a varying fraction of the group labels randomly flipped, and compare the performance of this to that of BARACK at an equivalent group prediction error rate. Results are in Figure 8; GDRO with the randomly perturbed group labels performs similarly to BARACK.

Specifically, we take the ground-truth group labels, randomly flip them to get the same error rate and confusion matrix as the predicted group labels from BARACK's group prediction model, and then use these "noisy group labels" as the groups for GDRO. For most settings, the final worst-group error is quite similar to that of BARACK, which suggests that the errors made by our group prediction model are indeed "sufficiently random" to not adversely affect the downstream worst-group accuracy too much. However, on U-MNIST, BARACK underperforms the random flipping version with a small number of

Table 3. Results with self-supervised pretrained model (RotNet), for 32 group-labeled examples per group as in Table 1. Worst-group accuracies are shown with and without the use of data augmentation.

Method	Waterbirds		CelebA		U-CIFAR10	
Worst-group Acc. (%)	No aug.	Aug.	No aug.	Aug.	No aug.	Aug.
ERM	38.4 ± 2.2	56.0 ± 3.8	39.3 ± 1.0	39.3 ± 2.1	79.3 ± 1.2	82.3 ± 2.3
BARACK-base	73.3 ± 0.6	83.0 ± 1.1	84.4 ± 1.5	84.6 ± 1.3	88.9 ± 1.5	89.7 ± 0.6
GDRO (full dataset)	79.6 ± 2.5	85.1 ± 0.8	82.6 ± 1.5	89.8 ± 0.5	93.6 ± 0.9	96.4 ± 0.4

group-labeled points, although this difference decreases as the number of group-labeled points increases.

Additional Details. As described above, for this experiment we created random "synthetic group pseudolabels" to have the same confusion matrix with respect to the true group labels as the confusion matrix of the BARACK group pseudolabels with respect to the true group labels. Specifically, for each setting (dataset, seed, and number of known group labels) we computed the confusion matrix of the group predictions output by the corresponding BARACK model (i.e., the one selected as the "best model" based on the criteria described in Appendix D.2), and then created the "synthetic pseudolabels" by taking the true group labels and flipping randomly chosen ones to match the confusion matrix. We then used these synthetic pseudolabels in the GDRO objective, with the same hyperparameters as those of the BARACK model. (Thus, one possible explanation for the fact that the BARACK models generally slightly outperform the "randomly flipped" models with equivalent error rates is the fact that we did not perform a separate hyperparameter search for the "randomly flipped" model, instead using the same ones as those of the BARACK model with the same confusion matrix.)

D.3.2. CLASS LABEL INPUT.

To evaluate the importance of using the *class* label as input in the first stage of BARACK, we run the standard BARACK procedure except without using the class label as input. This substantially decreases the final worst-group accuracy on all datasets (often by 10% or more, see Figure 8), showing that usage of the class label is indeed an important part of BARACK procedure. On most datasets, this drop can largely be explained by the reduced group prediction accuracy when the class feature is not used (the worst-group accuracy of the group classifier is up to 20 points worse when the class is not used). Interestingly, on U-CIFAR10, the group classifier's worst-group accuracy drops only 1-5% when it does not use the class label, but this still results in a drop of 8-11% in the worst-group accuracy of the final BARACK model.

Note that for all datasets considered, the classes are disjoint unions of the groups, meaning that knowing the class narrows down the possibilities for the group label. Thus, it is unsurprising that using the class label in the group prediction model significantly improves the worst-group accuracy of both the group prediction model and the final model. We hypothesize that the class label is more essential for group prediction in the spurious correlation setting (as suggested by the results in Fig. 8), because it enables the group prediction model to disambiguate between examples with the same spurious attribute but different classes, allowing it to focus on identifying the spurious attribute itself. By contrast, there is no clear "spurious attribute" on U-MNIST and U-CIFAR10; knowing the class label only reduces the number of candidate group labels for each example from 10 to 5.

D.3.3. PRETRAINED MODEL CHOICE.

As described in Section 3, on all tasks except U-MNIST, in our default experiments we start from a pretrained model trained on the supervised ImageNet task for all methods (and for both the group classifier and final model in BARACK). This is the standard approach when training on Waterbirds and CelebA, as in previous works (Sagawa et al., 2019; Sohoni et al., 2020; Levy et al., 2020; Liu et al., 2021). In this subsection, we investigate how the choice of pretrained model affects performance (Table 3). One motivation for this experiment is the potential for overlap or "leakage" between ImageNet and other image classification datasets (Kolesnikov et al., 2020). Moreover, we seek to confirm that the observations in previous sections regarding the performance of BARACK, compared to the baselines, are robust to the choice of pretrained model initialization. Thus, in this section we use pretrained models that were trained only with self-supervision (i.e., not using any labels). To be specific, in this section we use RotNet (Gidaris et al., 2018) from the VISSL library (Goyal et al., 2021) as opposed to the ResNet-50 pretrained on supervised ImageNet (from PyTorch) used in previous sections and previous works. Results are reported in Table 3 and plotted in Figure 9.

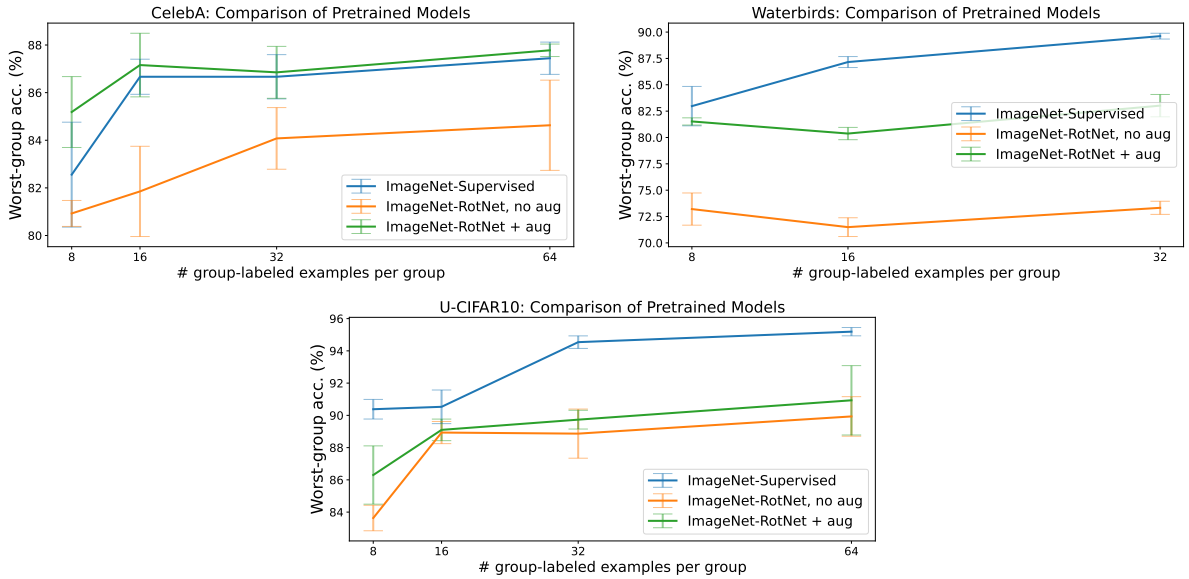

Figure 9. Worst-group accuracy using different initial pretrained models.

Overall, we observe that using the RotNet model achieves somewhat worse accuracies (both worst-group and average) for *all* methods, although this gap can be reduced or eliminated by using data augmentation. This unsurprisingly aligns with the findings of the original RotNet paper (Gidaris et al., 2018) that RotNet has somewhat worse transfer performance compared to supervised pretrained models. Despite the gap, the key takeaway is that BARACK still outperforms the baselines and remains competitive with full-dataset GDRO with the same initialization.

We additionally find that using data augmentation for the RotNet-based model can boost its performance closer to that of the supervised pretrained model. We observe that data augmentation does not seem to benefit the RotNet model much on U-CIFAR10, but it does on the other tasks (on CelebA, augmentation even boosts the performance using RotNet past the performance using the supervised ImageNet model, although of course the performance of the latter could likely also be boosted by using data augmentation). For the RotNet training with augmentation, the augmentations used in both Stage 1 and Stage 2 of BARACK are random crops, random flips, and random rotations (of up to 15 degrees). The reason we added the random rotations is because rotation is a key part of pretraining the RotNet model itself.

D.4. BARACK-SSL: Semi-Supervised Learning for Group Prediction

As an extension, we investigate the use of semi-supervised learning (SSL) using FixMatch (Sohn et al., 2020), for Stage 1 of BARACK. (We still use GDRO for Stage 2.) We refer to this procedure as BARACK-SSL. On the U-CIFAR10 task, BARACK-SSL can improve group prediction accuracy and, correspondingly, final robust performance. For example, with only 8 group-labeled examples per group, the worst-group accuracy of the final BARACK-SSL model is **94.0**%, compared to 90.4% for BARACK-Base. Correspondingly, the worst-group prediction accuracy of the group classifier is 83.7% when trained using FixMatch, while it is substantially lower at 42.9% when trained using simple supervised learning as in BARACK-Base, which helps explain these results. Thus, while BARACK-Base is simple and attains good worst-group accuracy, these results highlight the exciting potential of using more advanced SSL techniques to further boost worst-group performance of BARACK at the cost of more complexity. (Note: When training the group classifier using FixMatch, we use the *class* label as an input to the prediction head, just as in BARACK-Base.)

Additional Details. For BARACK-SSL, we use FixMatch (Sohn et al., 2020) to train the semi-supervised group classifier. We adapt the PyTorch implementation at `https://github.com/kekmodel/FixMatch-pytorch`, modifying it to use the class label the same way as described in Section 3 in order to assign zero probability to the groups that do not belong to the given class. Other than that, we use the default FixMatch hyperparameters. The group classifier used is simply the model at the end of FixMatch training (so we do not use the validation set at all for Stage 1). For the Stage 2 GDRO model, we used the same hyperparameter search and model selection approach as described in Appendix D.2.

Table 4. Extended version of Table 1, with additional baseline results (EIIL (Creager et al., 2021), Subset-GDRO) and results for BARACK when there are 8, 16, 32, and 64 group-labeled examples provided from each group (in each of the training and validation sets). As Waterbirds only has 56 training points in the smallest group, the last setting (64) does not apply to it.

Method	U-MNIST		Waterbirds		CelebA		U-CIFAR10	
Accuracy (%)	Worst-group	Avg.	Worst-group	Avg.	Worst-group	Avg.	Worst-group	Avg.
ERM	93.4 ± 0.5	99.2 ± 0.0	60.6 ± 3.3	97.3 ± 0.1	39.7 ± 3.0	95.7 ± 0.1	88.4 ± 1.4	99.5 ± 0.1
EIIL	97.2 ± 0.5	98.9 ± 0.2	87.3 ± 4.2	93.1 ± 0.6	81.3 ± 1.4	89.5 ± 0.4	85.3 ± 1.4	99.4 ± 0.1
GEORGE	95.7 ± 0.6	97.9 ± 0.2	76.2 ± 2.0	95.7 ± 0.5	53.7 ± 1.3	94.6 ± 0.2	93.4 ± 5.8	98.9 ± 0.3
JTT	96.2 ± 0.7	98.4 ± 0.4	88.0 ± 0.7	91.7 ± 0.8	77.8 ± 2.0	87.2 ± 1.2	89.0 ± 4.7	94.6 ± 1.3
Subset-GDRO (8)	66.9 ± 5.3	84.0 ± 1.8	76.4 ± 3.6	81.0 ± 4.9	56.9 ± 13.1	74.2 ± 5.3	83.4 ± 3.9	93.5 ± 0.6
Subset-GDRO (16)	77.4 ± 2.5	89.2 ± 1.1	83.9 ± 1.4	86.1 ± 1.7	75.5 ± 5.9	81.9 ± 1.4	86.0 ± 2.2	93.3 ± 1.8
Subset-GDRO (32)	85.4 ± 1.4	92.4 ± 0.4	86.9 ± 1.0	88.6 ± 0.5	76.6 ± 4.4	85.5 ± 1.8	88.6 ± 2.4	95.2 ± 0.9
Subset-GDRO (64)	89.6 ± 1.8	94.7 ± 0.7	-	-	79.0 ± 4.4	87.3 ± 1.2	91.7 ± 1.5	96.6 ± 0.4
Subset-GDRO (128)	92.8 ± 1.3	95.9 ± 0.3	-	-	81.4 ± 2.0	89.4 ± 0.4	94.6 ± 0.6	97.9 ± 0.1
BARACK-base (8)	96.2 ± 0.8	99.2 ± 0.1	83.0 ± 5.9	94.4 ± 2.8	81.1 ± 3.2	92.9 ± 0.3	90.4 ± 1.9	99.2 ± 0.2
BARACK-base (16)	96.4 ± 1.0	99.1 ± 0.2	86.9 ± 2.3	94.4 ± 2.9	83.0 ± 4.1	92.9 ± 0.8	90.5 ± 3.3	99.1 ± 0.4
BARACK-base (32)	96.9 ± 0.9	99.1 ± 0.3	89.6 ± 0.9	94.3 ± 1.3	83.8 ± 2.7	92.8 ± 0.6	94.5 ± 1.1	98.9 ± 0.3
BARACK-base (64)	97.4 ± 0.8	99.0 ± 0.3	-	-	84.3 ± 2.0	92.8 ± 0.5	95.2 ± 0.8	98.9 ± 0.4
BARACK-base (128)	97.5 ± 0.4	99.0 ± 0.2	-	-	87.0 ± 1.5	92.7 ± 0.4	96.5 ± 0.4	99.1 ± 0.2
Full-GDRO	98.6 ± 0.2	99.1 ± 0.1	90.9 ± 0.2	92.8 ± 0.2	89.3 ± 0.9	92.8 ± 0.1	97.0 ± 0.3	99.2 ± 0.3

D.5. Additional Results

D.5.1. WORST-GROUP PERFORMANCE.

In Table 4, we provide a more complete version of Table 1.

In this table, all results are rerun by us except for GEORGE on U-MNIST, CelebA and Waterbirds (since the original GEORGE paper already reports results over 5 seeds on those datasets). We ran 5 seeds for all methods except 10 seeds for BARACK, due to the increased variability from selecting different subsets of points with known group labels. We note that our averaged results were somewhat better on Waterbirds and worse on CelebA than those reported in the JTT paper (which reports results from one trial). Similarly, our results for EIIL on Waterbirds are also somewhat better than those reported in the EIIL paper (Creager et al., 2021).

Separately, the recent work (Zhang et al., 2022) proposes a method (CnC) based on contrastive learning for improving robustness to spurious correlations. This method does very well on the spurious correlation datasets (Waterbirds and CelebA), even exceeding the performance of BARACK on CelebA. However, we found that CnC did not work well on U-MNIST and U-CIFAR10 (attaining worst-group accuracies lower than those of ERM), which are tasks without explicit spurious correlations (even though CnC can in principle be applied to such tasks).

D.5.2. GROUP PREDICTION ACCURACY.

In Figure 10 we provide more results on the performance of the "Stage 1" group classification model. We plot both worst-group and average accuracies for the group classifier, corresponding to the same settings as in Table 4.

E. Conclusion

In this paper, our focus is to gain understanding into the practically important, yet so far understudied, version of the popular group robustness problem in which only a *few* group labels are known. We show simple techniques can be highly effective for this novel setting, and support our work with theoretical results and extensive ablations. Specifically, to address this problem, we present BARACK, a two-stage approach to improve group robustness in the setting when only a small number of group labels are known. We empirically validate that BARACK outperforms methods that do not use training group labels, even with just a small number of group-labeled examples. We theoretically provide generalization bounds on the worst-group performance of BARACK. Our results indicate that even a small number of group labels can be helpful for substantially improving worst-group performance.

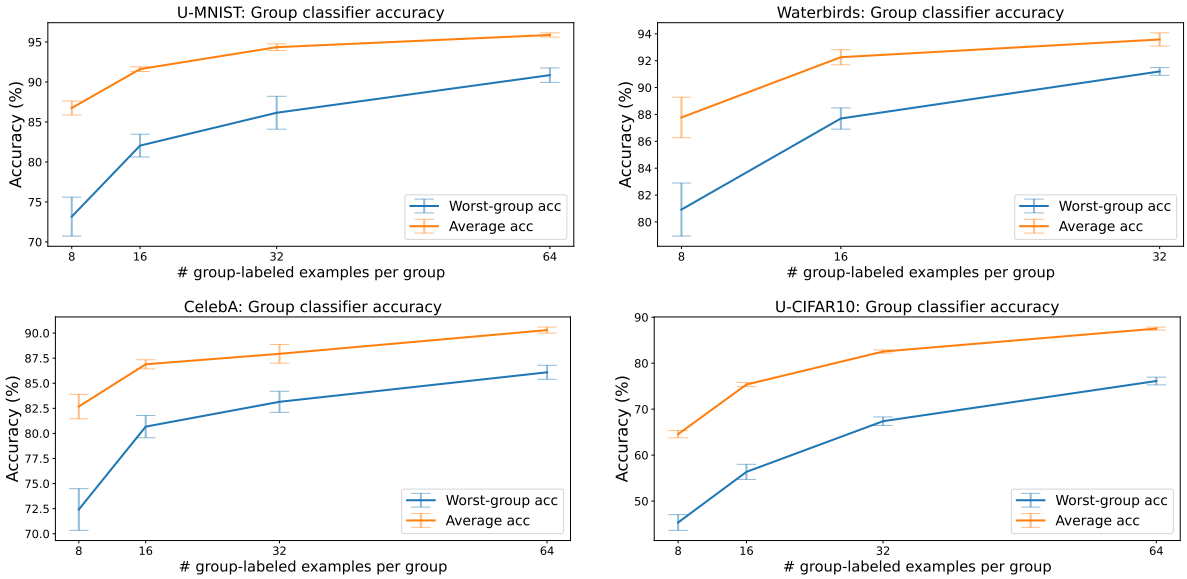

Figure 10. Worst-group and average accuracy of the *group* classifier.