Understanding Post-hoc Adaptation for Improving Subgroup Robustness

Anonymous Author(s)
Affiliation
Address
email

Abstract

A number of deep learning approaches have recently been proposed to improve model performance on subgroups under-represented in the training set. However, Menon et al. [13] recently showed that, models with poor subgroup performance can still learn representations which contain useful information about these subgroups. In this work, we explore the representations learned by various approaches to robust learning, finding that different approaches learn very similar representations. We probe a range of post-hoc procedures for making predictions from learned representations, showing that the distribution of the post-hoc validation set is paramount, and that clustering-based methods may be a promising approach.

1 Introduction

Machine learning systems trained with expected risk minimization (ERM) often struggle to perform well on under-represented subgroups in the training data [2, 5]. For this reason, a number of learning algorithms have been proposed to improve performance across subgroups, many taking advantage of subgroup information [3, 4, 8, 11]. However, it has also been noted that deep models trained with ERM can sometimes learn representations that contain sufficient information to perform well on all subgroups [13], even when the model's predictions yield large subgroup disparities.

In this work, we explore this apparent disconnect between representation learning and prediction in the context of group-robust classification. We consider a three-stage procedure, where a model is trained, then some post-hoc adaptation occurs directly on the learned representations, then tested for performance on various subgroups. We find that ERM-learned representations can be practically identical to those learned by more specialized methods that take advantage of subgroup information. Given this, we explore a range of procedures for post-hoc adaptation of a model by learning a new classifier on the representations directly, using a validation set and no subgroup information. We show that the distribution of the validation set is extremely important to obtaining good subgroup performance, and that clustering methods in representation space may be better than linear classifiers. Our results suggest that learning better representations for subgroup classification is a promising direction, and that post-hoc adaptation can be helpful for improving robustness.

2 Background

Notation. We assume a classification dataset of (input, label) pairs $\{(x_i, y_i)\}_{i=1}^n$, with $x \in \mathcal{X}, y \in \mathcal{Y} = \{1 \dots Y\}$. We may also have a categorical subgroup variable $\{g_i\}_{i=1}^n$ ($g \in \{1 \dots G\}$): $c_i = g \leftrightarrow$ example i is in subgroup g. Subgroup information is assumed available at training/validation time, but not test time. Usually, c will not be distributed uniformly throughout the training set — rather, some subgroups will be smaller, i.e. some values of c will be uncommon. Our goal is to learn a classification function $\bar{f} : \mathcal{X} \longrightarrow \mathcal{Y}$ which performs well on all subgroups. The metric by which we will evaluate our success is *worst-group accuracy*, that is, $\min_g \mathbb{E}[\mathbb{1}[\bar{f}(x) = y] | g_i = g]$. In this work, we assume $\bar{f}$ takes the form: $\bar{f} = \text{argmax} f(x); f(x) = \sigma(w^\top r(x) + b)$. Here, σ is the softmax function, w is a matrix containing a vector of weights for each class, b is a scalar bias, and r is a function outputting a vector representation of x. The model we use throughout, satisfying this functional form, is a Resnet-50 [7].

Train-Adapt-Test Procedure. Since we are interested in three aspects of model training (representation learning, post-hoc adaptation, and subgroup performance on test data), we consider a three-stage procedure. First, we initialize a model and **train** it on a training set with an unbalanced subgroup distribution; here, we learn the prediction function f and, as a byproduct, the representation function r. Next, we focus on r only, and perform post-hoc **adaptation**, using a validation set to learn a simple classifier on top of r. Finally, we **test** our adapted model (the composition of our post-hoc classifier and r) on held-out data, and record the performance on each subgroup.

Algorithms. We discuss several training algorithms, described fully in App. A.2. Expected risk minimization (**ERM**) is the usual paradigm for training ML models, where we ignore subgroup information and minimize mean loss on the training set. We also look at two **robust** approaches, which aim to take advantage of subgroup information g. The first is Group Distributionally Robust Optimization (**GDRO**) [19], which aims explicitly to minimize the worst group's average loss. The second is Invariant Risk Minimization (**IRM**) [1], uses a gradient penalty with the goal of learning a representation such that the same predictive classifier is optimal across subgroups.

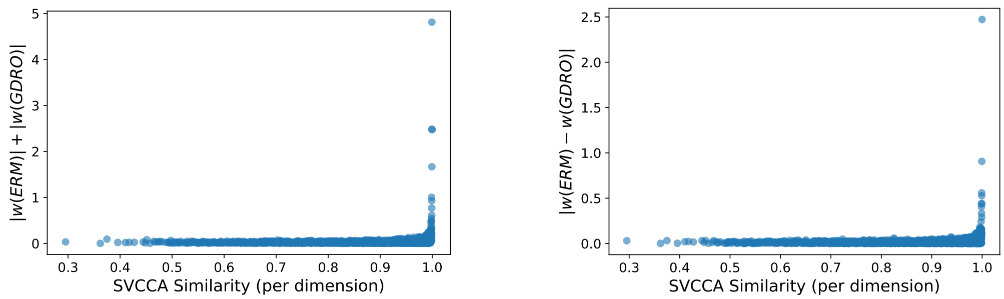
Dataset. We use the semi-synthetic Waterbirds dataset [19], which is created by pasting pictures of birds from CUB [21] onto backgrounds from Places-365 [22]. The task is to predict whether the bird in the image is a land or water bird; it is confounded by the background, which can be either land or water backgrounds. This yields four subgroups: land birds on land, water birds on water, land birds on water, and water birds on land. In the training set, there is a strong correlation between the bird and background factors: e.g. water birds are usually shown on water backgrounds. The dataset also contains a validation set, which is much more subgroup-balanced than the training set; the authors use the worst-subgroup accuracy on this validation set for early stopping. ERM obtains about 60% worst-group accuracy on Waterbirds, and GDRO/IRM obtain 87-90%. While subgroup classification may be harder on more realistic datasets, Waterbirds is a helpful tool to better understand the properties of various approaches in a research context. Since Waterbirds is small, we always initialize from a model pre-trained on Imagenet.

3 Feature Co-Discovery in Robust Learning

Deep learning approaches leveraging subgroup information g can improve worst-group performance [8, 9, 11, 12, 20]. We ask here: how do the learned representations reflect these more specialized approaches? Does using subgroup information at training time, or using more specialized algorithms, produce richer or better-separated learned features?

Prior literature suggests that differently-performing methods may nonetheless learn similar representations. For instance, Menon et al. [13] note that ERM features can be used to obtain similar subgroup performance to GDRO by learning a post-hoc linear model on a group-balanced validation set. This suggests that the necessary information needed to improve subgroup performance is already present in the ERM-learned representations, and that it is linearly extractible. In a meta-learning context, Raghu et al. [16] compares the representations learned by a model at its meta-initialization with the representations learned after performing task-specific adaptation. They find evidence of *feature re-use*: that the representations before and after task-specific adapation are similar, and the meta-initialized model and the task-adapted model differ mostly in their final classification heads.

In Fig. 2, we show the similarity of learned representations on Waterbirds across models trained with several loss functions. "None" is an Imagenet pre-trained model (not trained on Waterbirds); "ERM", "GDRO", and "IRM" use the methods from Sec. 2, initialized from the "None" model and trained (holding the random seed constant) on Waterbirds. "ERM-2" is the same as "ERM" with a different random seed. To compare the representations, we use SVCCA [17], which determines the most-aligned dimensions between the representations produced by two layers of neurons (here, the final layers of two different models) across some dataset (here, the Waterbirds validation set).



(a) The y-axis shows the sum of the absolute values in from the ERM and GDRO transformed classifiers. We see that the only features which are important for classification in the two models are co-discovered.

(b) The y-axis shows the absolute difference in the transformed classifier between ERM and GDRO models. We see the features which are treated highly differently between the two models are co-discovered.

Figure 1: Waterbirds feature co-discovery in ERM- and GDRO-learned representations. X-axis shows SVCCA similarity score for each dimension; a feature is co-discovered if it has a score near 1.

SVCCA returns a similarity score between 0 and 1 for each pair of (aligned) dimensions describing how "similar" they are between the two representations; these can be averaged to produce an overall similarity score which is "a direct multidimensional analogue of Pearson correlation", and describes the overall similarity of the two representations. Like Pearson correlation, closer to 1 is more similar.

Fig. 2 shows that ERM, GDRO and IRM all learn very similar representations on Waterbirds, with SVCCA overall similarities of ~ 0.9. This is comparable to the similarity between two different seeds of ERM, suggesting that the impact of these different algorithms on the representation is minimal here, despite methodological differences and the availability of subgroup information g to GDRO and IRM. Also, these methods are more similar to each other (~ 0.9) than to their initialization (~ 0.7). This explains the result from Menon et al. [13]: we can match GDRO performance using a post-hoc linear model on ERM representations because GDRO representations and ERM representations are roughly equal. Contrasted with Raghu et al. [16], we show that two models, using different learning algorithms, learned similar features to each other, but fairly different features from the initialized model; we call this *feature co-discovery* (rather than re-use [16]).

	ERM	GDRO	IRM	None	ERM-2
ERM	1	0.89	0.91	0.71	0.93
GDRO	0.89	1	0.97	0.72	0.88
IRM	0.91	0.97	1	0.73	0.9
None	0.71	0.72	0.73	1	0.71
ERM-2	0.93	0.88	0.9	0.71	1

Figure 2: SVCCA overall similarities between learned representations on Waterbirds. None is an Imagenet pre-trained model; ERM/GDRO/IRM are as in Sec. 2, with the same seed, using None as an initialization; ERM-2 is like ERM but with a different seed. 1 is perfect similarity.

We explore feature co-discovery between ERM and GDRO in Fig 1 (with similar ERM/IRM results in App. B). We use SVCCA to transform the representations from ERM and GDRO into their most similar directions, and obtain an equivalent linear classifier in transformed space for each method, such that the transformed representations and the transformed classifier together output the same logits as the non-transformed model. In Fig 1a, we plot the SVCCA similarity scores for each dimension on the x-axis. On the y-axis, we plot the sum of the absolute values of the transformed classifier weights from ERM and GDRO. We observe that all dimensions which are important for classification in either model are co-discovered (i.e. have high similarity score). In Fig 1b, we instead plot the absolute difference between the ERM and GDRO transformed classifiers on the y-axis. We observe that all dimensions where these two methods *differ* for classification are also co-discovered. Both of these plots suggest that the improvement in subgroup disparities on Waterbirds shown by GDRO is due to, not improvements in learned features, but a classification layer which weights the same features differently. See more experimental details for this section and the next in App. A.

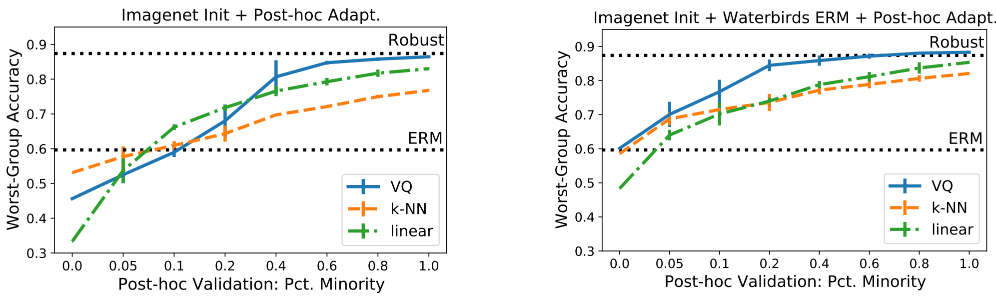


Figure 3: Waterbirds post-hoc classification results. On the left, we use the representations obtained by a a model pre-trained on Imagenet; on the right, we use the representations learned by ERM. On the x-axis, we show (number of minority examples) / (number of majority examples) in the post-hoc validation set. The y-axis shows the worst-group accuracy from the 4 subgroups on the test set. "Robust" shows the performance of GDRO/IRM, and "ERM" shows the performance of ERM; both are fully trained, starting from a pretrained Imagenet initialization. Average over 3 seeds shown.

4 Experiments: Post-hoc Classification

In Sec. 3, we show representations learned by ERM are similar to those learned by more specialized algorithms, including those using group information g. This supports the findings of Menon et al. [13], who show that by doing post-hoc logistic regression with a subgroup-balanced dataset, one can match the subgroup performance of GDRO. Here, we attempt to empirically disentangle two aspects of this finding, to determine which factor of the post-hoc learning procedure is important for subgroup performance. First is the post-hoc classification algorithm used: training a post-hoc linear model might induce larger disparities than a more flexible model would. Second is the distribution of the post-hoc validation set. As discussed previously, the given Waterbirds validation set is subgroup-balanced (P(background | label) = 0.5) — this is very different from the training set, and may encode a lot of information about the subgroups, even if only used post-hoc.

The three adaptation algorithms we look at are logistic regression, k-NN and vector quantization (VQ). In each, we train the post-hoc model on the representations in the validation set, and test them on representations from a second held-out set (the test set). For our VQ classifier: we fit a k-means model to each of the two classes on the validation set, returning two sets of k centroids, and use these 2k centroids in a 1-NN classifier for a given test point. To probe the importance of the validation distribution, we keep only $p\%$ of the minority examples in the validation set (minority examples have e.g. water bird on land background), and vary p. At $p = 1$, we have the original validation set — for each y, the number of land and water backgrounds are the same. At $p = 0$, we have removed all minority examples from the validation set. At $p \approx 0.05$, we have a validation set which is distributed similarly to the training set.

In Fig. 3, we show the effects of these two factors for the representations at the Imagenet pre-trained initialization, and the representations after ERM training on Waterbirds. We note that the distribution of the validation set is important, with a large difference in worst-group accuracy between the training set's proportion ($p = 0.05$) and the given validation set ($p = 1$). Indeed, post-hoc adaptation on the given validation set, using only a pre-trained model (which never sees any Waterbirds training data) matches the performance of specialized robust methods, which trains on the full Waterbirds training set as well as subgroup information, and might be considered an upper bound for post-hoc approaches. Secondly, the difference between the methods is smaller but noteworthy, and the difference between adaptation and the original model (horizontal line labelled "ERM") is large. Using the current validation set, VQ achieves the best performance, and this advantage is robust to perturbations in p: e.g. even at $p = 0.05$, VQ improves significantly over both the original model and linear adaptation.

Conclusion. We draw attention to the similarity of representations learned by robust approaches to those learned by ERM — this suggests that robust approaches which improve representation learning are potentially promising. On the other hand, the utility of post-hoc adaptation here stresses the richness of ERM representations, and it may be better to find methods which harness that richness through adaptation, rather than learning different ones.

References

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

[2] J. Buolamwini and T. Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pages 77–91. PMLR, 2018.

[3] E. Creager, J.-H. Jacobsen, and R. Zemel. Environment inference for invariant learning. In *ICML Workshop on Uncertainty and Robustness*, 2020.

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

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

[6] I. Gulrajani and D. Lopez-Paz. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020.

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

[8] C. Heinze-Deml and N. Meinshausen. Conditional variance penalties and domain shift robustness. *Machine Learning*, 110(2):303–348, 2021.

[9] W. Hu, G. Niu, I. Sato, and M. Sugiyama. Does distributionally robust supervised learning give robust classifiers? In *International Conference on Machine Learning*, pages 2029–2037. PMLR, 2018.

[10] J. Johnson, M. Douze, and H. Jégou. Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*, 2017.

[11] B. Kim, H. Kim, K. Kim, S. Kim, and J. Kim. Learning not to learn: Training deep neural networks with biased data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9012–9020, 2019.

[12] D. Krueger, E. Caballero, J.-H. Jacobsen, A. Zhang, J. Binas, D. Zhang, R. L. Priol, and A. Courville. Out-of-distribution generalization via risk extrapolation (rex). *arXiv preprint arXiv:2003.00688*, 2020.

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

[14] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019.

[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[16] A. Raghu, M. Raghu, S. Bengio, and O. Vinyals. Rapid learning or feature reuse? towards understanding the effectiveness of maml. *arXiv preprint arXiv:1909.09157*, 2019.

[17] M. Raghu, J. Gilmer, J. Yosinski, and J. Sohl-Dickstein. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. *arXiv preprint arXiv:1706.05806*, 2017.

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

[19] S. Sagawa, P. W. Koh, T. B. Hashimoto, and P. Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

[20] M. Srivastava, T. Hashimoto, and P. Liang. Robustness to spurious correlations via human annotations. In *International Conference on Machine Learning*, pages 9109–9119. PMLR, 2020.

[21] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

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

A Experimental Details

A.1 Model Training

We train (where not otherwise indicated) using SGD with learning rate of 1e-4, momentum of 0.9, L2 weight regularization of 1e-4, and batch size 128. We used early stopping with patience 20 for all models, using an early stopping metric of worst-group accuracy on the validation set for all models except those for the coloured lines in Fig. 3, which use reweighted validation loss as an early stopping metric (i.e. validation loss where the subgroup losses are reweighted to match the training distribution's subgroup distribution). We used $K = 5$ for GDRO and $\lambda = 3$ for IRM. We train all models in Pytorch [14] using their Imagenet-pretrained initialization [18].

A.2 Algorithms

We define here a number of different sueprvised classification algorithms of interest. We let ℓ be example-wise cross-entropy. Let the number of examples in a group g be n_g, and let the following shorthand describe the average loss for a group g: $\ell_g(f) = \frac{1}{n_g} \sum_{i=1}^{n} \ell(f(x_i), y_i) \mathbb{1}\{g_i = g\}$.

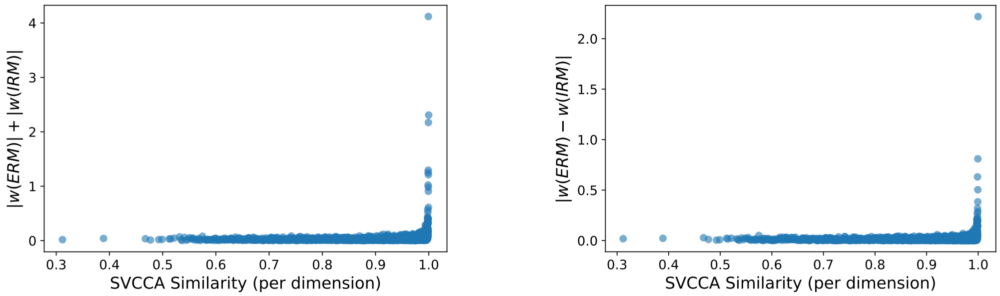
Expected Risk Minimization Expected risk minimization (ERM) is the usual paradigm for training ML models. In ERM, we minimize the mean loss ℓ on the training set: $\mathcal{L}_{ERM}(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i)$.

Group DRO. Group Distributionally Robust Optimization (GDRO) [19] was proposed specifically for subgroup-based learning, and takes advantage of the subgroup information g. This loss aims to ensure that no group's loss is that bad, and the group adjustment term (with hyperparameter K) ensures greater focus on smaller groups, which may otherwise be ignored: $\mathcal{L}_{GDRO}(f) = \max_{g \in \{1 \ldots G\}} \left\{ \ell_g(f) + \frac{K}{\sqrt{n_g}} \right\}$.

IRM. Invariant risk minimization (IRM) [1] is another method that uses subgroup information. The intuition for this is somewhat involved [1]; the overarching motivation is that each environment should learn a representation such that the same predictive classifier is optimal across environments. The second term below is a gradient penalty on the output of f, w is a constant multiplier on the output of f, and λ is a hyperparameter: $\mathcal{L}_{IRM}(f) = \sum_{g=1}^{G} \ell_g(f) + \lambda \|\nabla_{w|w=1} \ell_g(w \cdot f)\|$

A.3 Post-hoc Training

We used the implementations of k-NN and k-Means implemented in the `faiss` package [10]. We used the implementation of logistic regression implemented in the `scikit-learn` package [15] with the "lbfgs" solver. For each method, we searched over 5 hyperparameter values and chose the best one (by worst-group error on the test set) for each value reported, as suggested by Gulrajani and Lopez-Paz [6]. For VQ and k-NN, we loop over the values of $k = 1, 2, 4, 8, 16$. For logistic regression, we loop over the L2-regularization value $C = 0.1, 0.2, 0.5, 1.0, 2.0$.



(a) The y-axis shows the sum of the absolute values in from the ERM and IRM transformed classifiers. We see that the only features which are important for classification in the two models are co-discovered.

(b) The y-axis shows the absolute difference in the transformed classifier between ERM and IRM models. We see the features which are treated highly differently between the two models are co-discovered.

Figure 4: Waterbirds feature co-discovery analysis on ERM- and IRM-learned representations. The x-axis shows the SVCCA similarity score for each dimension in both plots; a feature is co-discovered if it has a score near 1.

B Feature Co-Discovery in ERM and IRM

In Fig. 1, we show evidence of feature co-discovery between ERM and GDRO. Since GDRO and IRM have very similar representations (as shown in Fig. 2), we would expect a similar pattern to hold in IRM's representations. For completeness, we show the analogous results for ERM and IRM in Fig. 4.