Evaluating and Improving Robustness of Self-Supervised Representations to Spurious Correlations

Kimia Hamidieh [1 2] **Haoran Zhang** [3] **Marzyeh Ghassemi** [3 2]

Abstract

Recent empirical studies have found inductive biases in supervised learning toward simple features that may be spuriously correlated with the label, resulting in suboptimal performance on minority subgroups. Despite the growing popularity of methods which learn representations from unlabeled data, it is unclear how potential spurious features may be manifested in the learnt representations. In this work, we explore whether recent Self-Supervised Learning (SSL) methods would produce representations which exhibit similar behaviors under spurious correlation. First, we show that classical approaches in combating spurious correlations, such as dataset re-sampling during SSL, do not consistently lead to invariant representation. Second, we find that spurious information is represented disproportionately heavily in the later layers of the encoder. Motivated by these findings, we propose a method to remove spurious information from these representations during pretraining, by pruning or re-initializing later layers of the encoder. We find that our method produces representations which outperform the baseline on three datasets, without the need for group or label information during SSL.

1. Introduction

Many real-world predictive tasks contain spurious correlations – features that are correlated with the label only for certain subsets of the data (Shah et al., 2020; McCoy et al., 2019; Gururangan et al., 2018). In such cases, where such correlations are easier to learn than the label itself (Nagarajan et al., 2020; Sagawa et al., 2020b), Empirical Risk Minimization (ERM) models have been shown to make predictions based on spurious correlations, leading to systematically poor performance for minority subgroups (Hashimoto et al., 2018). For instance, models trained to detect pneumonia (Zech et al., 2018) or COVID-19 (DeGrave et al., 2021) from chest X-rays across hospitals use information about the source hospital as a shortcut for patient disease, rather than invariant pulmonary characteristics.

There have been many methods proposed to tackle spurious correlations for supervised learning. Some methods require group information during training (Sagawa et al., 2020a), while others do not (Liu et al., 2021a; Zhang et al., 2022). However, all methods require group information for model selection. The problem becomes more complex when we do not have access to label information. For instance, unlabeled data with underlying latent features that could exhibit such correlations with downstream labels may be used for pre-text tasks, and biases caused by these correlations may propagate to downstream tasks.

Recently, there has been a rise in the popularity of Self-Supervised Learning (SSL) methods, which seek to learn data representations from large, *unlabeled* datasets (Chen et al., 2020a; He et al., 2019; Grill et al., 2020; Chen & He, 2020; Caron et al., 2020; Zbontar et al., 2021). In many real-world applications, important underlying features of the data that are necessary to be captured in the representations for downstream tasks, may be correlated with simpler features unrelated to these tasks. We suspect that since SSL models can discriminate data samples in the majority groups using the simpler features, they may rely mostly on these features for the instance discrimination pre-text task. As such, it is unclear how spurious correlations are reflected in representations that SSL models learn.

In this work, we consider a simple setting where we are interested in capturing one *core* feature in representations, which is correlated with a *spurious*, simpler feature. In this setting, we ask the following questions: (1) Do SSL methods also have a bias towards learning representations which result in spurious correlations when a linear ERM model is trained on representations for downstream tasks? (2) If so, can we suppress such features from learnt representations during SSL, without access to any group or label information?

[1] University of Toronto [2] Vector Institute [3] Massachusetts Institute of Technology. Correspondence to: Kimia Hamidieh <kimia@cs.toronto.edu>.

Published at the ICML 2022 Workshop on Spurious Correlations, Invariance, and Stability. Baltimore, Maryland, USA. Copyright 2022 by the author(s).

We make the following contributions:

- We show that simple techniques for avoiding spurious correlations during supervised learning, such as reweighting (Idrissi et al., 2021) or re-sampling (Sagawa et al., 2020b) of the training set with group information, does not necessarily improve representations learnt with SSL.
- We show that group information is correlated with features learned in the final layers of the network, and hypothesize that removing such information from the final layers can be beneficial.
- We propose a method to eliminate spurious feature information from representations learnt during SSL while maintaining discriminative ability for downstream predictive tasks, without access to group or label information.

2. Related Work

Spurious Correlations Spurious correlations arise in supervised learning models in a variety of domains, from medical imaging (Zech et al., 2018; DeGrave et al., 2021) to natural language processing (Tu et al., 2020; Wang & Culotta, 2020). A variety of approaches have been proposed to learn classifiers which do not make use of spurious information. Methods like GroupDRO (Sagawa et al., 2020a) and DFR (Kirichenko et al., 2022) require group information during training, while methods like JTT (Liu et al., 2021a), LfF (Nam et al., 2020), CVaR DRO (Duchi et al., 2019), CnC (Zhang et al., 2022) do not. However, all methods require group information for model selection.

Self-supervised Representation Learning Self-supervised learning methods learn representations from large-scale unlabeled datasets where annotations are scarce. In vision applications, the pretext task is typically to maximize similarity between two augmented views of the same image (Jing & Tian, 2020). This can be done in a contrastive fashion using the InfoNCE loss (Oord et al., 2018), as in SimCLR (Chen et al., 2020a) and MoCo (Chen et al., 2020b), or without the need for negative samples at all, as in BYOL (Grill et al., 2020), SwAV (Caron et al., 2020), SimSiam (Chen & He, 2021), and Barlow Twins (Zbontar et al., 2021).

Learning under Dataset Imbalance and Shortcuts Self-supervised models have been found to be more robust to dataset imbalance (Liu et al., 2021b), and Jiang et al. (2021b;a); Liu et al. (2021b) further address the subgroup gaps. Robinson et al. (2021) addressed shortcut learning in contrastive learning by adversarially modifying encoded features. Other works in addressing group robustness or fairness in SSL, however, require group information or labels (Tsai et al., 2020; Song et al., 2019; Wang et al., 2021). In the supervised setting, how subnetworks of a trained model can affect minority examples (Hooker et al., 2019) or out-of-distribution generalization (Zhang et al., 2021), and forgetting features via final-layer re-initialization (Zhou et al., 2022), have also been studied.

For a more comprehensive summary of the background and related work, see Appendix A.

3. Methods

3.1. Problem Setup

We suppose that data is generated from underlying feature space $\mathcal{Z} = \{z_{\text{core}}, z_{\text{spur}}, \dots\}$, where z_{core} and z_{spur} are correlated for unlabeled data available for pre-text task, and z_{core} determines labels y for our downstream task of interest, while z_{spur} determines the spurious attribute, which is easier to learn, and is not of interest of downstream tasks. Our goal, is to be able to predict y from the learned representations in the downstream task where such correlations do not hold.

3.2. Investigating the Extent of Spurious Learning in SSL

First, we design three experiments to establish the extent of spurious learning in SSL, and how easily it could be removed by simple solutions.

Baseline Spurious SSL: We first empirically evaluate learning of the core and spurious features on self-supervised representations. We train SimSiam (Chen & He, 2020) models with ResNet-18 backbones on 4 datasets containing spurious correlations (See Appendix B for dataset descriptions). Then, we evaluate the learned representations using a balanced dataset which we create by subsampling majority groups (Sagawa et al., 2020b; Idrissi et al., 2021), to avoid the statistical and geometrical skews (Nagarajan et al., 2020) when training the linear classifier on representations. We use the balanced dataset to separately train two linear classifiers to predict labels and spurious attribute.

Spurious Feature Removal Effectiveness using Group Information: In the next step, we examine whether classical approaches for combating spurious correlations, such as re-sampling training examples, are effective in removing spurious information during SSL. Assuming that group information is available, we train SimSiam on datasets resampled using the following strategies: (i) downsampling examples in majority groups to have the same number of examples in all groups, (ii) upsampling minority examples to have the same number of examples in all groups.

Investigating Spurious Signals in Layer-Wise Feature Representations: We also investigate methods that identify the sources of spurious correlation, tailored to the SSL setting. One crucial observation from prior work is that during

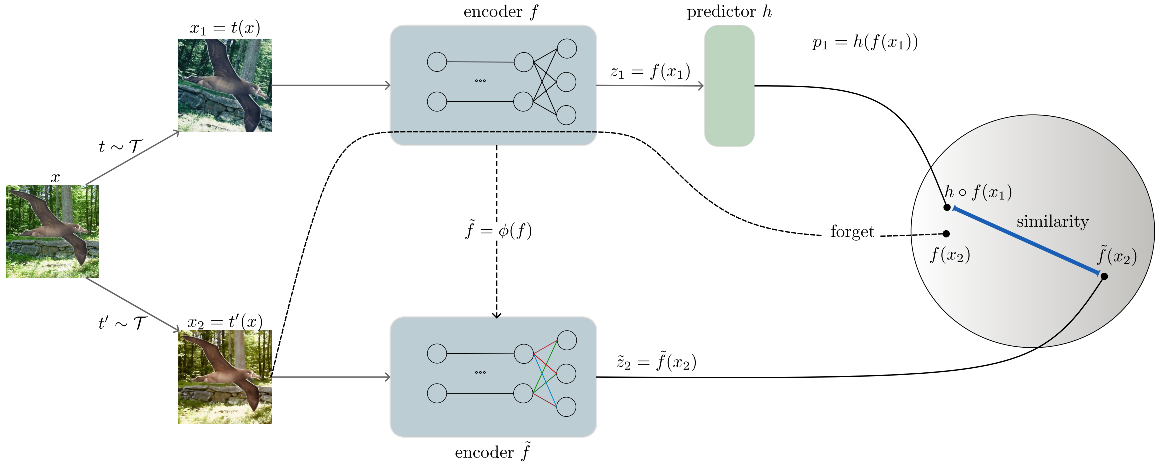

Figure 1: We use model transformation modules to create new views of training examples in the feature space. The introduced set of transformations removes the features learned in the final few layers, and final representations are invariant to such transformations.

supervised learning, overparameterized models have an inductive bias towards learning the spurious feature and "memorizing" the minority examples, even after re-sampling or reweighting (Sagawa et al., 2020b). It has been also observed in the supervised setting that memorization predominately occurs in the deeper layers (Zhang et al., 2019; Stephenson et al., 2021) and that the representations in earlier layers of the network correctly classify the easier examples while the final layers memorize the difficult examples (Baldock et al., 2021).

We suspect that a similar phenomenon happens during SSL, as the spurious feature is easier to infer, and almost sufficient for the instance discrimination task. To verify this hypothesis, we evaluate the mutual information between feature representations across the layers of the trained encoder with (1) the labels $I(Z;Y)$, and (2) the spurious attribute $I(Z;G)$.

3.3. Late-layer Transformation-based View Generation

Motivated by these experiments, we define *Late*-layer *t*ransformation-based *v*iew *g*eneration modules – *Late-TVG*.

Assuming that z_{spur} is easier to learn, and majority examples can be discriminated by this feature as well as z_{core}, the core feature may not be fully captured in SSL representations. However, if the positive pairs have the same z_{core} but different or removed z_{spur}, the encoder must learn the core features (Robinson et al., 2021). Motivated by how SSL models are invariant to the set of image augmentations imposed by the augmentation module, we propose a model transformation module, that specifically targets modifying the spurious feature in encoded examples, in order to improve core feature representation.

More formally, we consider a model transformation module $\mathcal{U}$, that transforms any given function f_θ parameterized by $\theta = \{W_1, \ldots, W_n\}$ to $f_{\tilde{\theta}}$. At each step, we draw a transformation $\phi_{M,\theta'}$ form $\mathcal{U}$ to obtain transformed encoder $f_{\tilde{\theta}}$ from f_θ. Each model transformation can be defined with a mask $M \in \{0,1\}^{|\theta|} = \{M^1, \ldots, M^n\}$, where we reparameterize the unmasked weights $(1-M) \odot \theta'$ (either reinitialize or set them to 0), and keep the rest of the weights $M \odot \theta$ the same, thus $\tilde{\theta} = \phi_{M,\theta'}(\theta) = M \odot \theta + (1-M) \odot \theta'$.

Transformations: In our experiments, we consider two types of transformation modules as below:

- Re-initialization of the final-layers (**SSL$_{\textbf{Reinit, L}}$**): Re-initializing the weights in layers deeper than L. $\mathcal{U}_{\text{Reinit, L}} = \{\phi_{M_L, \theta_{\text{Reinit}}} \mid \theta_{\text{Reinit}} \sim \mathcal{D}_\theta\}$ where M_L is masking all weights before layer L or $M_L = \{M_L^l \mid M_L^l = \mathbb{1}(l < L)^{|W_l|}, l \in [n]\}$, and $\mathcal{D}_\theta$ is the parameter initialization distribution.

- Threshold Pruning (**SSL$_{\textbf{Prune, L, a}}$**): Magnitude pruning $a\%$ of the weights in all layers deeper than L. $\mathcal{U}_{\text{Prune, L, a}} = \{\phi_{M_{L,a}, \theta_0}; \theta_0 = (0)^{|\theta|}\}$ where $M_{L,a} = \{M_L^l \odot \text{Top}_a(W_l) \mid l \in [n]\}$ and $\text{Top}_a(W_l)_{i,j} = \mathbb{I}(W_{l(i,j)}$ in top $a\%$ of $\theta)$

Note that these transformation modules are designed to avoid memorization and *forget* representations of the minority examples, by removing the undesired information in the final layers, and we want the output of the transformed module to be close to the original encoded examples in the representation space. In other words, the model should be invariant to model transformations in the feature space, and thus these transformations can be considered as curated view generating operations for the minority groups. This encourages the encoder to be invariant to final layer transfor-

		kNN (Target)		LR (Target)		LR (Group)
Dataset	SSL Training	Average	Worst Group	Average	Worst Group	Average
`waterbirds`	Normal	53.4%	49.7%	56.2%	48.0%	87.4%
	Downsample	51.7%	47.0%	55.5%	50.5%	85.3%
	Upsample	66.3%	9.5%	62.3%	46.7%	81.5%
`metashift`	Normal	54.2%	28.8%	66.6%	27.6%	64.9%
	Downsample	45.1%	0.0%	59.8%	0.0%	57.4%
	Upsample	56.5%	32.9%	65.8%	24.2%	65.1%
`spurcifar10`	Normal	57.1%	22.1%	60.3%	36.2%	52.0%
	Downsample	36.0%	19.6%	48.8%	28.8%	48.3%
	Upsample	43.8%	20.7%	53.6%	9.3%	70.7%
`cmnist`	Normal	86.7%	42.0%	83.7%	37.6%	85.9%
	Downsample	86.2%	23.7%	80.0%	35.2%	83.2%
	Upsample	86.9%	39.4%	80.0%	43.7%	79.6%

Table 1: Accuracy of ResNet-18 encoders trained using SSL on four datasets, evaluated on a balanced test set. We vary the SSL training set used (Normal: original dataset, Downsample: downsampling majority groups, Upsample: upsampling minority groups). Re-sampling does not necessarily improve worst-group accuracy in the downstream task.

mations, and thus helps discriminating minority examples based on the core feature.

To learn these representations, given two random augmentations $t, t' \sim \mathcal{T}$ from the augmentation module $\mathcal{T}$, two views $x_1 = t(x)$ and $x_2 = t'(x)$ are generated from an input image x. At each step, given a feature encoder f, and an augmentation module $\mathcal{U}$, we obtain a transformed model $\tilde{f} = \phi(f)$, $\phi \sim \mathcal{U}$. During training, one example x_1 is passed through the normal encoder $v_1 = f(x_1)$, and the other example x_2 is passed through the transformed encoder $\tilde{v}_2 = \tilde{f}(x_2)$. Encoded feature $\tilde{v}_2$ is now a positive example that should be close to v_1 in the representation space. In our experiments, we use SimSiam in which the encoder f aims to maximize the cosine similarity of the two views via predictor network h, and a stop-gradient operator as in Figure 1. Given previously defined features v_1 and $\tilde{v}_2$, the cosine similarity $\mathcal{D}(h(v_1), \texttt{stopgrad}(\tilde{v}_2))$ will be maximized at each step.

Experimental Setup: We train SimSiam models with proposed transformed modules. For simplicity, we use one module with a fixed pruning percentage and layer threshold throughout training. At each epoch t, given a transformation from the fixed module is applied to encoder f_t to generate the transformed model $\tilde{f}_t$. The model transformations are applied to the branch of SimSiam that a gradient-stop operation is applied to later. We use group information in the validation set and measure worst-group accuracy in order to choose layer threshold and pruning percentage hyperparameters (In Appendix D we show that even an uncurated set of model transformations enhance worst-group performance), and evaluate them similar to previous experiments.

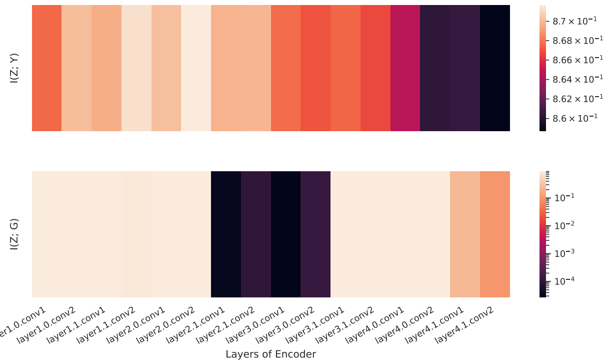

Figure 2: Comparison of Mutual Information between features and labels (top) vs. spurious attribute and labels (bottom) across layers of a ResNet-18 model trained with SimSiam on coloured MNIST. We observe that $I(Z;G)$ decreases in the intermediate layers, and grows back in the final layers, indicating that the final representations rely on the spurious feature for the instance discrimination task in SSL.

4. Results

4.1. SSL Suffers From Spurious Correlations

From Table 1, we find that across all datasets, SSL models exhibit gaps between worst-group and average accuracy when predicting the core feature, indicating that even when spurious correlation does not hold for downstream tasks, the learnt features are more predictive of the spurious feature in comparison to the core one. This is in contrast with supervised learning (Rosenfeld et al., 2022), where such

		kNN (Target)		LR (Target)		LR (Group)	Other Metrics		
Dataset	Method	Worst Group	Average	Worst Group	Average	Average	I(Z; Y)	I(Z; G)	L_{align}
cmnist	SSL_{Base}	41.91%	86.72%	37.61%	83.67%	85.92%	8.56E-01	5.40E-05	1.148
	$\text{SSL}_{\text{Reinit, 15}}$	33.90%	86.19%	42.37%	78.89%	79.72%	8.42E-01	8.89E-01	0.511
	$\text{SSL}_{\text{Prune, 0.7, 19}}$	37.04%	88.49%	50.00%	81.62%	82.07%	8.60E-01	9.03E-01	1.21
	$\text{SSL}_{\text{Prune, 0.9, 18}}$	32.20%	89.04%	44.07%	84.32%	81.55%	8.54E-01	9.02E-01	1.12
	$\text{SSL}_{\text{Prune, 0.99, 19}}$	35.19%	88.31%	44.07%	79.64%	82.00%	8.51E-01	8.73E-01	0.947
metashift	SSL_{Base}	28.82%	54.17%	27.68%	66.57%	65.00%	1.07E-01	6.17E-02	0.848
	$\text{SSL}_{\text{Reinit, 18}}$	21.38%	61.23%	18.64%	62.95%	66.52%	1.54E-03	1.86E-01	0.940
	$\text{SSL}_{\text{Prune, 0.7, 17}}$	27.59%	60.09%	18.64%	63.09%	67.24%	1.31E-01	1.89E-01	0.958
	$\text{SSL}_{\text{Prune, 0.8, 17}}$	24.83%	58.66%	20.34%	64.95%	64.66%	7.37E-02	1.58E-01	0.955
	$\text{SSL}_{\text{Prune, 0.9, 18}}$	25.52%	59.23%	18.64%	64.66%	65.52%	3.85E-02	1.46E-01	0.947
spurcifar10	SSL_{Base}	22.12%	57.05%	36.24%	60.36%	52.02%	1.96E+00	3.43E-03	1.072
	$\text{SSL}_{\text{Prune, 0.3, 18}}$	31.25%	61.60%	45.83%	66.86%	48.91%	1.98E+00	1.10E-05	1.08
	$\text{SSL}_{\text{Prune, 0.5, 18}}$	30.77%	58.44%	43.75%	66.23%	49.37%	2.06E+00	1.98E-09	1.05
	$\text{SSL}_{\text{Prune, 0.7, 19}}$	35.42%	58.06%	41.67%	65.80%	49.79%	2.03E+00	2.30E-02	1.04
	$\text{SSL}_{\text{Prune, 0.9, 17}}$	25.00%	50.98%	39.58%	58.59%	48.66%	1.65E+00	5.56E-05	0.630
waterbirds	SSL_{Base}	49.72%	53.40%	48.04%	56.18%	87.45%	8.63E-02	5.84E-01	0.867
	$\text{SSL}_{\text{Prune, 0.3, 17}}$	48.29%	50.59%	52.34%	55.59%	86.54%	5.03E-02	5.92E-01	0.858
	$\text{SSL}_{\text{Prune, 0.7, 17}}$	43.77%	54.99%	51.56%	62.10%	81.76%	7.86E-02	5.72E-01	0.873
	$\text{SSL}_{\text{Prune, 0.8, 18}}$	35.83%	60.23%	50.78%	60.10%	84.10%	1.01E-01	4.60E-01	0.802
	$\text{SSL}_{\text{Prune, 0.9, 17}}$	47.54%	50.88%	52.18%	57.82%	89.80%	9.63E-02	0.595E-01	0.702

Table 2: Top model transformations for each dataset; The learned representations in each case, we freeze the representations to evaluate the representations with: (i) average and worst-group of a 5-NN classifier (ii) Average and worst-group of a linear classifier (iii) Average accuracy of a linear classifier trained to infer the spurious feature (iv) Mutual information between representations and classes (v) Mutual information between representations and spurious feature (vi) Alignment loss (Zhang et al., 2022) which is indicator of how close examples within the same class are in the representation space. In many cases, the worst-group accuracy of linear classifier is improved; in other cases kNN accuracy has improvements;

models contain enough core information to perform well on all subgroups, and only needing a re-training of the final layer on a balanced validation set. (Menon et al., 2021).

4.2. Re-sampling During SSL is Not Useful

From Table 1, we observe that re-sampling during self-supervised training does not improve downstream worst-group accuracy. Given that the downstream linear model is trained on a down-sampled dataset where such correlations do not exist, this means that re-sampling during self-supervised training does not necessarily improve linear separability of representations with respect to the core feature, even in balanced datasets.

4.3. Layer-Wise Feature Representations

In Figure 2, we see that spurious features are disproportionately represented in later layers of the network, while invariant features are represented throughout the network. This confirms our hypothesis that later layers contain more spurious information during SSL, and motivates our proposed method.

4.4. Transformation-based Disentanglement in Final Layers Improves Worst-group Performance

As shown in Table 2, *Late-TVG* improves the worst-group accuracy of the linear classifier is improved by up to more than 10% on `spurcifar10`. However, it does not seem to improve the worst-group accuracy in `metashift`. We suspect that this is due to the spurious feature (outdoor vs. indoor) being more difficult to infer than the invariant feature, which violates the assumptions of our method.

5. Conclusion

We studied the performance of SSL models when trained on spuriously correlated data, and showed that the core feature is not very well distinguished in many cases. We further examined spurious learning in SSL models. Finally, we proposed a method - *Late-TVG* - that improves the worst-group downstream performance of SSL models without having access to group or label information, and empirically validated its performance on four datasets. Future work include benchmarking other SSL methods such as SimCLR (Chen et al., 2020a), and adapting the method to other modalities such as natural language (Tu et al., 2020), and considering multiple core features in the underlying feature space.

References

Adel, T., Valera, I., Ghahramani, Z., and Weller, A. One-network adversarial fairness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 2412–2420, 2019.

Alemi, A., Poole, B., Fischer, I., Dillon, J., Saurous, R. A., and Murphy, K. Fixing a broken elbo. In *International Conference on Machine Learning*, pp. 159–168. PMLR, 2018.

Amini, A., Soleimany, A. P., Schwarting, W., Bhatia, S. N., and Rus, D. Uncovering and mitigating algorithmic bias through learned latent structure. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 289–295, 2019.

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

Baldock, R., Maennel, H., and Neyshabur, B. Deep learning through the lens of example difficulty. *Advances in Neural Information Processing Systems*, 34, 2021.

Beutel, A., Chen, J., Zhao, Z., and Chi, E. H. Data decisions and theoretical implications when adversarially learning fair representations. *arXiv preprint arXiv:1707.00075*, 2017.

Burgess, C. P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., and Lerchner, A. Understanding disentangling in β-vae. *arXiv preprint arXiv:1804.03599*, 2018.

Cai, R., Li, Z., Wei, P., Qiao, J., Zhang, K., and Hao, Z. Learning disentangled semantic representation for domain adaptation. In *IJCAI: proceedings of the conference*, volume 2019, pp. 2060. NIH Public Access, 2019.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020.

Chen, R. T., Li, X., Grosse, R., and Duvenaud, D. Isolating sources of disentanglement in variational autoencoders. *arXiv preprint arXiv:1802.04942*, 2018.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020a.

Chen, X. and He, K. Exploring Simple Siamese Representation Learning. *arXiv e-prints*, art. arXiv:2011.10566, November 2020.

Chen, X. and He, K. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15750–15758, 2021.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems*, 29, 2016.

Chen, X., Fan, H., Girshick, R., and He, K. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020b.

DeGrave, A. J., Janizek, J. D., and Lee, S.-I. Ai for radiographic covid-19 detection selects shortcuts over signal. *Nature Machine Intelligence*, 3(7):610–619, 2021.

Duchi, J. C., Hashimoto, T., and Namkoong, H. Distributionally robust losses against mixture covariate shifts. *Under review*, 2, 2019.

Eastwood, C. and Williams, C. K. A framework for the quantitative evaluation of disentangled representations. In *International Conference on Learning Representations*, 2018.

Edwards, H. and Storkey, A. Censoring representations with an adversary. *arXiv preprint arXiv:1511.05897*, 2015.

Esmaeili, B., Wu, H., Jain, S., Bozkurt, A., Siddharth, N., Paige, B., Brooks, D. H., Dy, J., and Meent, J.-W. Structured disentangled representations. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2525–2534. PMLR, 2019.

Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B., and Smola, A. A kernel method for the two-sample-problem. *Advances in neural information processing systems*, 19: 513–520, 2006.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z. D., Gheshlaghi Azar, M., Piot, B., Kavukcuoglu, K., Munos, R., and Valko, M. Bootstrap your own latent: A new approach to self-supervised Learning. *arXiv e-prints*, art. arXiv:2006.07733, June 2020.

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

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

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. *arXiv:1911.05722*, 2019.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016.

Hooker, S., Courville, A., Clark, G., Dauphin, Y., and Frome, A. What do compressed deep neural networks forget? *arXiv preprint arXiv:1911.05248*, 2019.

Hsieh, J.-T., Liu, B., Huang, D.-A., Fei-Fei, L. F., and Niebles, J. C. Learning to decompose and disentangle representations for video prediction. *Advances in neural information processing systems*, 31, 2018.

Idrissi, B. Y., Arjovsky, M., Pezeshki, M., and Lopez-Paz, D. Simple data balancing achieves competitive worst-group-accuracy. *arXiv preprint arXiv:2110.14503*, 2021.

Jiang, Z., Chen, T., Chen, T., and Wang, Z. Improving contrastive learning on imbalanced data via open-world sampling. *Advances in Neural Information Processing Systems*, 34, 2021a.

Jiang, Z., Chen, T., Mortazavi, B., and Wang, Z. Self-damaging contrastive learning. *arXiv preprint arXiv:2106.02990*, 2021b.

Jing, L. and Tian, Y. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(11): 4037–4058, 2020.

Kim, H. and Mnih, A. Disentangling by factorising. In *International Conference on Machine Learning*, pp. 2649–2658. PMLR, 2018.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kirichenko, P., Izmailov, P., and Wilson, A. G. Last layer re-training is sufficient for robustness to spurious correlations. *arXiv preprint arXiv:2204.02937*, 2022.

Liang, W. and Zou, J. Metashift: A dataset of datasets for evaluating contextual distribution shifts and training conflicts. *arXiv preprint arXiv:2202.06523*, 2022.

Lin, Z., Thekumparampil, K., Fanti, G., and Oh, S. Infogan-cr and modelcentrality: Self-supervised model training and selection for disentangling gans. In *international conference on machine learning*, pp. 6127–6139. PMLR, 2020.

Liu, E. Z., 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. In *International Conference on Machine Learning*, pp. 6781–6792. PMLR, 2021a.

Liu, H., HaoChen, J. Z., Gaidon, A., and Ma, T. Self-supervised learning is more robust to dataset imbalance. *arXiv preprint arXiv:2110.05025*, 2021b.

Louizos, C., Swersky, K., Li, Y., Welling, M., and Zemel, R. The variational fair autoencoder. *arXiv preprint arXiv:1511.00830*, 2015.

Madras, D., Creager, E., Pitassi, T., and Zemel, R. Learning adversarially fair and transferable representations. In *International Conference on Machine Learning*, pp. 3384–3393. PMLR, 2018.

McCoy, R. T., Pavlick, E., and Linzen, T. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. *arXiv preprint arXiv:1902.01007*, 2019.

Menon, A. K., Rawat, A. S., and Kumar, S. Overparameterisation and worst-case generalisation: friend or foe? In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=jphnJNOwe36.

Nagarajan, V., Andreassen, A., and Neyshabur, B. Understanding the failure modes of out-of-distribution generalization. *arXiv preprint arXiv:2010.15775*, 2020.

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

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

Ridgeway, K. and Mozer, M. C. Learning deep disentangled embeddings with the f-statistic loss. *Advances in neural information processing systems*, 31, 2018.

Robinson, J., Sun, L., Yu, K., Batmanghelich, K., Jegelka, S., and Sra, S. Can contrastive learning avoid shortcut solutions? *arXiv preprint arXiv:2106.11230*, 2021.

Rosenfeld, E., Ravikumar, P., and Risteski, A. Domain-adjusted regression or: Erm may already learn features sufficient for out-of-distribution generalization. *arXiv preprint arXiv:2202.06856*, 2022.

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, 2020a.

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

Shah, H., Tamuly, K., Raghunathan, A., Jain, P., and Netrapalli, P. The pitfalls of simplicity bias in neural networks. *Advances in Neural Information Processing Systems*, 33: 9573–9585, 2020.

Song, J., Kalluri, P., Grover, A., Zhao, S., and Ermon, S. Learning controllable fair representations. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2164–2173. PMLR, 2019.

Stephenson, C., Padhy, S., Ganesh, A., Hui, Y., Tang, H., and Chung, S. On the geometry of generalization and memorization in deep neural networks. *arXiv preprint arXiv:2105.14602*, 2021.

Tsai, Y.-H. H., Wu, Y., Salakhutdinov, R., and Morency, L.-P. Demystifying self-supervised learning: An information-theoretical framework. *arXiv e-prints*, pp. arXiv–2006, 2020.

Tu, L., Lalwani, G., Gella, S., and He, H. An empirical study on robustness to spurious correlations using pre-trained language models. *Transactions of the Association for Computational Linguistics*, 8:621–633, 2020.

Wang, T., Yue, Z., Huang, J., Sun, Q., and Zhang, H. Self-supervised learning disentangled group representation as feature. *Advances in Neural Information Processing Systems*, 34:18225–18240, 2021.

Wang, Z. and Culotta, A. Identifying spurious correlations for robust text classification. *arXiv preprint arXiv:2010.02458*, 2020.

Xie, Q., Dai, Z., Du, Y., Hovy, E., and Neubig, G. Controllable invariance through adversarial feature learning. *arXiv preprint arXiv:1705.11122*, 2017.

Zbontar, J., Jing, L., Misra, I., LeCun, Y., and Deny, S. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pp. 12310–12320. PMLR, 2021.

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

Zemel, R., Wu, Y., Swersky, K., Pitassi, T., and Dwork, C. Learning fair representations. In *International conference on machine learning*, pp. 325–333. PMLR, 2013.

Zhang, B. H., Lemoine, B., and Mitchell, M. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 335–340, 2018.

Zhang, C., Bengio, S., Hardt, M., Mozer, M. C., and Singer, Y. Identity crisis: Memorization and generalization under extreme overparameterization. *arXiv preprint arXiv:1902.04698*, 2019.

Zhang, D., Ahuja, K., Xu, Y., Wang, Y., and Courville, A. Can subnetwork structure be the key to out-of-distribution generalization? In *International Conference on Machine Learning*, pp. 12356–12367. PMLR, 2021.

Zhang, M., Sohoni, N. S., Zhang, H. R., Finn, C., and Ré, C. Correct-n-contrast: A contrastive approach for improving robustness to spurious correlations. *arXiv preprint arXiv:2203.01517*, 2022.

Zhao, H. and Gordon, G. J. Inherent tradeoffs in learning fair representations. *arXiv preprint arXiv:1906.08386*, 2019.

Zhou, H., Vani, A., Larochelle, H., and Courville, A. Fortuitous forgetting in connectionist networks. *arXiv preprint arXiv:2202.00155*, 2022.

Zhu, Y., Min, M. R., Kadav, A., and Graf, H. P. S3vae: Self-supervised sequential vae for representation disentanglement and data generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6538–6547, 2020.

A. Related Work and Background

A.1. Group Robustness

Empirical risk minimization (ERM) minimizes the average training loss across training points. Given a loss function $\ell(x, y; \theta)$, ERM minimizes the following objective:

$$J_{\text{ERM}}(\theta) = \frac{1}{n} \sum_{i=1}^{n} \ell(x_i, y_i; \theta). \tag{1}$$

Group distributionally robust optimization (Group DRO) uses training group information to minimize the worst-group error on the training set, assuming we have access to group annotations on the training data $\{(x_1, y_1, g_1), \ldots (x_n, y_n, g_n)\}$. Given a loss function $\ell(x, y; \theta)$, the objective can then be written as:

$$J_{\text{groupDRO}}(\theta) = \max_{g \in \mathcal{G}} \frac{1}{n_g} \sum_{i|g_i=g} \ell(x_i, y_i; \theta) \tag{2}$$

where n_g is the number of training points with group $g_i = g$.

Just Train Twice (JTT) is a simple two-stage approach that does not require group annotations at training time. First, it trains an identification model $\hat{f}_{\text{id}}$ via ERM and then identifies an error set $E = \{(x_i, y_i)$ s.t. $\hat{f}_{\text{id}}(x_i) \neq y_i\}$ of training examples that $\hat{f}_{\text{id}}$ misclassifies. Then, it trains a final model $\hat{f}_{\text{final}}$ by upweighting the points in the identified error set.

$$J_{\text{up-ERM}}(\theta, E) = \left(\lambda_{\text{up}} \sum_{(x,y) \in E} \ell(x, y; \theta) + \sum_{(x,y) \notin E} \ell(x, y; \theta) \right), \tag{3}$$

Correct-n-Contrast (CnC) learns an idendification model similar to JTT to identify samples with the same class but dissimilar spurious features, and then trains a model with contrastive learning to learn similar representations for same-class samples. More percisely, it jointly trains the model's encoder layers f_{enc} with a contrastive loss and the full model f_θ with a cross-entropy loss with the following objective:

$$\hat{\mathcal{L}}\left(f_\theta; x, y\right) = \lambda \hat{\mathcal{L}}_{\text{con}}^{\text{sup}}\left(f_{\text{enc}}; x, y\right) + (1 - \lambda) \hat{\mathcal{L}}_{\text{cross}}\left(f_\theta; x, y\right).$$

Where $\hat{\mathcal{L}}_{\text{con}}^{\text{sup}}\left(f_{\text{enc}}; x, y\right)$ is the supervised contrastive loss of x and its positive and negative samples, based on whether the identifier model has made a mistake on samples or no, and $\hat{\mathcal{L}}_{\text{cross}}\left(f_\theta; x, y\right)$ is average cross-entropy loss over x, the M positives, and N negatives, and λ is a balancing hyperparameter.

A.2. Self-supervised Representation Learning

Self-supervised representation learning methods learn visual representations from large-scale unlabeled images where data annotations are scarce and time-consuming. Contrastive learning is a discriminative approach to learn representations that aims to attract similar or positive samples and push apart different or negative samples, which has become increasingly successful in recent years (Chen et al., 2020a; He et al., 2019; Grill et al., 2020; Caron et al., 2020; Zbontar et al., 2021). The standard approach for generating positive pairs without additional annotations is to create multiple views of each data point using random augmentations. The contrastive learning loss or InfoNCE (Oord et al., 2018) then maximizes a lower bound on the mutual information between the two views.

For instance, SimCLR (Chen et al., 2020a) generates two randomly augmented views of each image $\tilde{\boldsymbol{x}}_i = t(\boldsymbol{x}), \tilde{\boldsymbol{x}}_j = t'(\boldsymbol{x}), \quad t, t' \sim \mathcal{T}$ given a batch of images, and uses all other augmentated samples from the batch as negative examples. Then it uses an encoder f to extract representations from these augmented examples, and a small projection head g which maps these representations to the contrastive loss space. Given a minibatch of N samples, the InfoNCE loss is optimized for the sum of all examples in the minibatch.

Some proposed methods discard the need for negative samples in contrastive learning. BYOL (Grill et al., 2020) uses a siamese architecture with momentum encoders to prevent different representations from collapsing into one vector. SwAV (Caron et al., 2020) exploits online clustering for each batch to enforce consistency between cluster assignments from different views, and SimSiam (Chen & He, 2020) uses a simple stop-gradient operation in a siamese architecture to avoid collapsing.

We use SimSiam (Chen & He, 2020) in particular in our experiments. Similar to SimCLR it creates two randomly augmented views x_1 and x_2 from an image x. Then it uses encoder f consisting of a backbone such as ResNet and a projection MLP head to create representations of the two views. A prediction MLP head h, transforms the output of one view and matches it to the other view. Given two output vectors are $p_1 \triangleq h(f(x_1))$ and $z_2 \triangleq f(x_2)$; SimSiam minimizes the negative cosine similarity $\mathcal{D}(p_1, z_2)$:

$$\mathcal{D}(p_1, z_2) = -\frac{p_1}{\|p_1\|_2} \cdot \frac{z_2}{\|z_2\|_2},$$

where $\|\cdot\|_2$ is ℓ_2-norm. Then they define a symmetrized loss for each image with a stop-gradient operator to avoid collapse as below:

$$\mathcal{L} = \frac{1}{2}\mathcal{D}(p_1, \texttt{stopgrad}(z_2)) + \frac{1}{2}\mathcal{D}(p_2, \texttt{stopgrad}(z_1)).$$

Note that we use the word encoder to address the backbone in f, since projection layers are thrown away when evaluating the representations.

A.3. Disentangled Representations

A similar line of research is creating representations where each dimension is independent and corresponds to a particular attribute (Eastwood & Williams, 2018; Ridgeway & Mozer, 2018), some works study learning such representations in a supervised manner (Hsieh et al., 2018; Cai et al., 2019) while unsupervised approaches rely on VAEs (Higgins et al., 2016; Chen et al., 2018; Zhu et al., 2020) and GANs (Chen et al., 2016; Lin et al., 2020).

The works in fair representation learning usually address removing sensitive attributes from the representations by: obfuscating any information about sensitive attributes in order to approximately satisfy demographic parity (Zemel et al., 2013), using adversarial methods (Edwards & Storkey, 2015; Xie et al., 2017; Beutel et al., 2017; Zhang et al., 2018; Madras et al., 2018; Adel et al., 2019; Zhao & Gordon, 2019), or feature disentanglement based using variational approaches. (Kingma & Welling, 2013; Gretton et al., 2006; Louizos et al., 2015; Amini et al., 2019; Alemi et al., 2018; Burgess et al., 2018; Chen et al., 2018; Kim & Mnih, 2018; Esmaeili et al., 2019).

Perhaps the closest to our work is (Wang et al., 2021) where samples are partitioned into two subsets that correspond to an entangled group element followed by minimizing a subset-invariant contrastive loss, where the invariance guarantees to disentangle the group element.

B. Datasets

We make use of the following four image datasets:

- `waterbirds` (Sagawa et al., 2020a): Background (land, water) is spuriously correlated with bird type (labdbird, waterbird)
- `cmnist` (Colored MNIST) (Arjovsky et al., 2019): Color of digit on the images spuriously correlated with the binary class based on the number
- `spurcifar10` (Spurious CIFAR10) (Nagarajan et al., 2020): Color of lines on the images spuriously correlated with the class
- `metashift` (Liang & Zou, 2022): Cats vs Dogs task: Background (indoor, outdoor) spuriously correlated with pet type (cat, dog)

C. Comparing SSL to CLIP representations

We train linear classifiers with different re-sampled sets of training examples on frozen CLIP (Radford et al., 2021) representations. These representations have found to be more robust to distribution shifts, and we aim to answer if balanced downstream training set can improve worst-group accuracy. As shown in table 3, even CLIP representations do not help mitigate the geometrical and statistical skews when learning the linear classifier on frozen representations.

		k-NN		Linear probe		Spurious Attribute (Linear)	
dataset	Linear Train set	Average	Worst-group	Average	Worst-group	Average	Worst-group
CelebA	Normal	83.51%	20.39%	89.67%	16.98%	77.09%	61.26%
	Downsample	78.57%	75.56%	88.83%	81.11%	97.04%	91.11%
	Upsample	90.89%	28.33%	91.76%	86.11%	98.57%	90.56%
Waterbirds	Normal	56.77%	30.53%	61.30%	45.08%	68.06%	45.09%
	Downsample	69.23%	63.81%	79.01%	74.01%	89.59%	86.30%
	Upsample	74.06%	43.93%	75.35%	55.61%	83.26%	68.85%
Metashift	Normal	74.44%	50.85%	76.06%	7.91%	63.19%	22.60%
	Downsample	84.69%	68.97%	80.69%	30.51%	73.68%	22.03%
	Upsample	88.41%	73.79%	82.55%	32.20%	75.54%	35.59%
Colored MNIST	Normal	73.59%	54.65%	72.64%	55.57%	74.14%	70.40%
	Downsample	91.84%	89.39%	89.04%	85.98%	98.08%	96.21%
	Upsample	97.18%	62.88%	93.76%	88.26%	97.95%	97.16%
Spurious CIFAR10	Normal	25.11%	14.58%	35.79%	22.87%	93.62%	0.00%
	Downsample	33.20%	10.42%	52.56%	33.33%	51.29%	30.00%
	Upsample	39.28%	20.83%	58.17%	41.67%	58.14%	35.42%

Table 3: Average and worst-group test accuracy of CLIP representations trained on Normal, Downsample, Upsampletrain sets. Balancing the training set used for linear evaluation helps us identify the learned representations by avoiding the statistical and geometrical skews

D. Additional Representation Augmentation Results

Even without having access to group information in the validation set, untuned hyperparameters (layer threshold and pruning percentage), we improve worst-group accuracy in most cases.

		kNN (Target)		LR (Target)		LR (Group)	Other Metrics		
Dataset	Method	Worst Group	Average	Worst Group	Average	Average	I(Z; Y)	I(Z; G)	L_{align}
cmnist	SSL_{Base}	41.9%	86.7%	37.6%	83.6%	85.9%	8.56E-01	5.40E-05	1.148
	$\text{SSL}_{\text{Reinit, 17}}$	39.0%	81.2%	35.6%	87.8%	81.3%	8.48E-01	8.76E-01	0.620
	$\text{SSL}_{\text{Reinit, 19}}$	35.6%	80.3%	37.3%	89.0%	82.0%	8.57E-01	9.05E-01	0.875
	$\text{SSL}_{\text{Thres, 0.7}}$	37.3%	84.3%	39.0%	88.8%	83.2%	8.58E-01	9.06E-01	1.235
	$\text{SSL}_{\text{Thres, 0.95}}$	33.3%	84.6%	31.5%	90.9%	84.3%	8.53E-01	7.19E-01	1.141
metashift	SSL_{Base}	28.8%	54.1%	27.6%	66.5%	65.0%	1.07E-01	6.17E-02	0.848
	$\text{SSL}_{\text{Reinit, 17}}$	16.9%	62.8%	26.2%	57.2%	64.5%	7.37E-06	9.31E-02	0.906
	$\text{SSL}_{\text{Reinit, 19}}$	15.3%	67.0%	23.4%	60.8%	65.2%	1.19E-02	1.42E-01	0.942
	$\text{SSL}_{\text{Thres, 0.7}}$	15.3%	64.7%	22.8%	61.4%	65.2%	1.05E-02	1.69E-01	0.971
	$\text{SSL}_{\text{Thres, 0.95}}$	11.9%	62.8%	27.6%	58.5%	64.9%	5.21E-09	3.05E-02	0.924
spurcifar10	SSL_{Base}	22.1%	57.0%	26.2%	60.4%	52.0%	1.96E+00	3.43E-03	1.072
	$\text{SSL}_{\text{Reinit, 17}}$	31.7%	51.5%	25.8%	48.8%	50.2%	1.28E+00	4.63E-03	0.333
	$\text{SSL}_{\text{Reinit, 19}}$	31.3%	59.3%	22.8%	54.8%	49.5%	1.83E+00	6.86E-10	0.644
	$\text{SSL}_{\text{Thres, 0.7}}$	31.3%	57.7%	29.2%	51.5%	47.9%	1.88E+00	8.12E-11	0.946
	$\text{SSL}_{\text{Thres, 0.95}}$	38.5%	59.0%	30.6%	54.0%	48.6%	1.89E+00	1.47E-10	0.894
waterbirds	SSL_{Base}	49.7%	53.4%	48.0%	56.2%	87.4%	8.63E-02	5.84E-01	0.867
	$\text{SSL}_{\text{Reinit, 17}}$	45.5%	50.0%	46.8%	51.6%	84.5%	5.78E-02	4.31E-01	0.282
	$\text{SSL}_{\text{Reinit, 19}}$	46.7%	55.2%	22.7%	55.6%	88.3%	5.12E-02	5.82E-01	0.392
	$\text{SSL}_{\text{Thres, 0.7}}$	51.6%	62.1%	43.8%	55.0%	81.8%	7.86E-02	5.72E-01	0.873
	$\text{SSL}_{\text{Thres, 0.95}}$	42.2%	58.0%	38.5%	53.6%	87.1%	9.69E-02	5.39E-01	0.684

Table 4: The same set of augmentations for each dataset; without using group annotations of validation data, worst-group accuracy can be improved.