Exploring Spurious Learning in Self-Supervised Representations

Anonymous Author(s)
Affiliation
Address
email

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 [1, 2, 3]. For instance, models trained to detect pneumonia [4] or COVID-19 [5], from chest X-rays across hospitals use a *spurious* underlying feature – information about the source hospital – as a shortcut for predicting the pathology, rather than invariant pulmonary characteristics which are the *core* or *invariant* features. In cases where spurious correlations are easier to learn than invariant correlations [6, 7], Empirical Risk Minimization (ERM) models have been shown to make predictions based on spurious correlations. These models have systematically poor performance for minority subgroups where such correlations do not hold [8]. Learning to mitigate spurious shortcuts is well explored in *supervised learning* with targeted solutions like importance weighting [9], re-sampling [10, 11], or approaches based on group distributionally robust optimization [12].

Spurious learning becomes more complex when such correlations appear in unlabeled data. Self-Supervised Learning (SSL) methods aim to learn representations from unlabeled datasets through solving an auxiliary pretext tasks [13]. For instance, recently proposed SSL algorithms learn representations [14, 15, 16, 17, 18, 19] by discriminating instances within the training set [20]. In this context, important underlying features of the data, e.g., different identifiers of the disease, should be captured in the representations for downstream tasks rather than a simple proxy. For instance, in a setting where we have unlabeled x-rays from multiple hospital sites, we may want to perform SSL initially on this unlabeled data, and use the learnt representations for a downstream task like infection prediction. During SSL the spurious feature (hospital) could be correlated with core feature (infection present in x-ray), *and* be more easily learnt for pre-text tasks. However, this spurious feature is not invariantly useful for determining infection status – our intended downstream task. In this setting, biases embedded in latent correlations propagate to downstream tasks where the

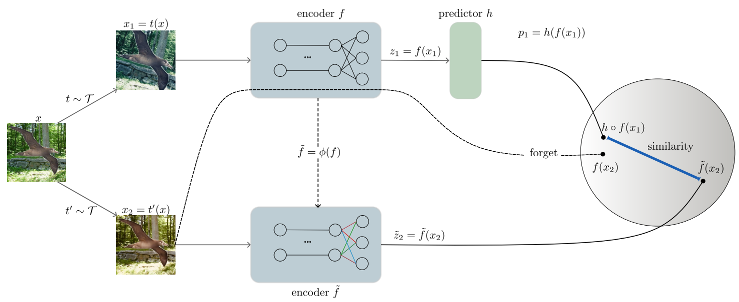

Figure 1: We use model transformation modules to create new views of training examples in the representation space. The introduced set of transformations removes the features learned in the final few layers, and final representations are invariant to such transformations.

association is spurious, e.g, while there are more infected patients in large hospitals, infected patients should still be classified as infected in all hospitals. Importantly, we cannot directly penalize this spurious learning in vanilla SSL as the data is unlabeled, and the final downstream task is unspecified.

In this work we address the open question of how spurious correlations are reflected in the representations that SSL models learn. We consider a simple setting where we are interested in capturing one *core* feature in the representations, which is correlated with a *spurious*, simpler feature. Inspired by a crucial observations from supervised learning on overparameterized models, we target a method that addresses two potential issues. First, overparameterized models have an inductive bias towards learning the spurious feature and “memorizing" the minority examples, even after re-weighting [10] or re-sampling [7]. Second, memorization predominately occurs in the deeper layers [21, 22] where earlier layers of the network correctly classify the easier examples while the final layers memorize the difficult examples [23].

We propose *model transformations* or *Late-TVG* - a method that induces invariance to spurious feature in the representation space by eliminating the memorization of minority groups. We run experiments in different settings on four datasets, and find that we are able to maintain discriminative ability for downstream predictive tasks, without access to group or label information. Our work makes the following contributions:

- We find that known techniques for avoiding spurious correlations during supervised learning, such as re-weighting or re-sampling of the training set with group information, does not consistently improve representations learnt with SSL.
- We show that minority groups are predicted by features learnt in the final layers of SSL networks, and hypothesize that regularizing the final layers improves learning of the more complex, or core, features [24].
- We find that *Late-TVG* effectively improves worst-group performance in downstream tasks in four datasets by enforcing core feature learning.

2 Methods

We assume that data is generated from underlying latent 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.

Motivated by improved SSL model invariance when trained with augmentations in *image* space [14], we propose a model transformation module that specifically targets augmentations that modify the spurious feature in *representation* space. We propose **L**ate-layer **T**ransformation-based View Generation - *Late-TVG* , which uses transformations to overcome spurious learning in SSL models and improve core feature representation.

2.1 Late-layer Transformation-based View Generation

Formally, we consider a model transformation module $\mathcal{U}$, that transforms any given function f_θ parameterized by $\theta = \{W_1, \dots, W_n\}$ 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 $M \in \{0,1\}^{|\theta|} = \{M^1, \dots, M^n\}$, where we re-parameterize 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 (**$\mathbf{SSL}_{\textbf{Reinit, L}}$**): 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 (**$\mathbf{SSL}_{\textbf{Prune, L, a}}$**): 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 $\text{Top}_a(W_l)_{i,j} = \mathbb{I}(W_{l(i,j)} \text{ in top } a\% \text{ of } \theta)$

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 and x_2 are respectively passed through the normal encoder $v_1 = f(x_1)$, and 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 (see Appendix D).

Similarly to Image Augmentation modules inducing representation space invariance based on visual similarities, View Generation modules encourage the encoder to be invariant to final layer transformations. We hypothesize that this enables learning more complex features in the earlier layers of the encoder.

3 Experiments

3.1 Investigating the Extent of Spurious Learning in SSL

We design three experiments to establish the extent of spurious learning in SSL, how easily it can be removed by simple solutions, and the impact of using *Late-TVG*.

Baseline Evaluation of Spurious Learning in SSL: We first empirically evaluate learning of the core and spurious features on self-supervised representations. Furthermore, we compare features of a *supervised* model trained by ERM, to self-supervised representations.

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 [10], are effective in removing spurious information during SSL. Assuming that group information is available, we train SimSiam on datasets re-sampled 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 design experiments to verify that spurious features are easier to extract than core features in SSL, and are sufficient for the instance discrimination task. To do so, 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 $I(Z;G)$.

3.1.1 Experimental Setup

We investigate the performance of *Late-TVG* on four commonly used datasets containing spurious correlations – CMNIST [25], MetaShift [26], Spurious CIFAR-10 [6], and Waterbirds [27] (See Appendix F for dataset descriptions). We train SimSiam [17] models with ResNet-18 backbones on these datasets which contain spurious correlations. Then, we evaluate the learned representations using a balanced dataset where the correlation does not hold. To create the downstream training dataset, we subsample majority groups [7, 10], to avoid the statistical and geometrical skews [6] when of the linear classifier on representations. In each case we report the average and worst-group accuracy of downstream logistic regression (LR) and k-nearest neighbors (KNN) classifiers. For completeness, we also report the mutual information between the representations and the label ($I(Z;Y)$) and group variables ($I(Z;G)$), as well as the alignment loss [28]. We compare our method against standard SSL without model transformations (SSL_{Base}).

4 Results

SSL Suffers From Spurious Correlations. From Table 2, 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 [29], where such 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. [30].

Re-sampling During SSL is Not Useful. From Table 2, 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.

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.

Transformation-based Disentanglement in Final Layers Improves Worst-group Performance. As shown in Table 1, *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.

		kNN (Target)		**LR (Target)**		**LR (Group)**	**Other Metrics**		
Dataset	**Method**	**Worst Group**	**Average**	**Worst Group**	**Average**	**Average**	**I(Z; Y)**	**I(Z; G)**	$\mathbf{L_{align}}$
`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 1: 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 [28] 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;

5 Conclusion

We proposed a method - *Late-TVG* - that improves the worst-group downstream performance of SSL models on spuriously correlated data without having access to group or label information, and empirically validated its performance. Future work include benchmarking other SSL methods such as SimCLR [14], and adapting the method to other modalities such as natural language [11], and considering multiple core features in the underlying feature space.

References

[1] Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. *Advances in Neural Information Processing Systems*, 33:9573–9585, 2020.

[2] R Thomas McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. *arXiv preprint arXiv:1902.01007*, 2019.

[3] Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R Bowman, and Noah A Smith. Annotation artifacts in natural language inference data. *arXiv preprint arXiv:1803.02324*, 2018.

[4] John R Zech, Marcus A Badgeley, Manway Liu, Anthony B Costa, Joseph J Titano, and Eric Karl Oermann. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: a cross-sectional study. *PLoS medicine*, 15(11):e1002683, 2018.

[5] Alex J DeGrave, Joseph D Janizek, and Su-In Lee. Ai for radiographic covid-19 detection selects shortcuts over signal. *Nature Machine Intelligence*, 3(7):610–619, 2021.

[6] Vaishnavh Nagarajan, Anders Andreassen, and Behnam Neyshabur. Understanding the failure modes of out-of-distribution generalization. *arXiv preprint arXiv:2010.15775*, 2020.

[7] Shiori Sagawa, Aditi Raghunathan, Pang Wei Koh, and Percy Liang. An investigation of why overparameterization exacerbates spurious correlations. In *International Conference on Machine Learning*, pages 8346–8356. PMLR, 2020.

[8] Tatsunori Hashimoto, Megha Srivastava, Hongseok Namkoong, and Percy Liang. Fairness without demographics in repeated loss minimization. In *International Conference on Machine Learning*, pages 1929–1938. PMLR, 2018.

[9] Preethi Lahoti, Alex Beutel, Jilin Chen, Kang Lee, Flavien Prost, Nithum Thain, Xuezhi Wang, and Ed H. Chi. Fairness without demographics through adversarially reweighted learning, 2020.

[10] Badr Youbi Idrissi, Martin Arjovsky, Mohammad Pezeshki, and David Lopez-Paz. Simple data balancing achieves competitive worst-group-accuracy. *arXiv preprint arXiv:2110.14503*, 2021.

[11] Lifu Tu, Garima Lalwani, Spandana Gella, and He He. An empirical study on robustness to spurious correlations using pre-trained language models. *Transactions of the Association for Computational Linguistics*, 8:621–633, 2020.

[12] Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization, 2020.

[13] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430, 2015.

[14] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[15] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv:1911.05722*, 2019.

[16] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised Learning. *arXiv e-prints*, page arXiv:2006.07733, June 2020.

[17] Xinlei Chen and Kaiming He. Exploring Simple Siamese Representation Learning. *arXiv e-prints*, page arXiv:2011.10566, November 2020.

[18] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020.

[19] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR, 2021.

[20] Zhirong Wu, Yuanjun Xiong, Stella Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance-level discrimination. *arXiv:1805.01978*, 2018.

[21] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Michael C Mozer, and Yoram Singer. Identity crisis: Memorization and generalization under extreme overparameterization. *arXiv preprint arXiv:1902.04698*, 2019.

[22] Cory Stephenson, Suchismita Padhy, Abhinav Ganesh, Yue Hui, Hanlin Tang, and SueYeon Chung. On the geometry of generalization and memorization in deep neural networks. *arXiv preprint arXiv:2105.14602*, 2021.

[23] Robert Baldock, Hartmut Maennel, and Behnam Neyshabur. Deep learning through the lens of example difficulty. *Advances in Neural Information Processing Systems*, 34, 2021.

[24] Samira Abnar, Mostafa Dehghani, Behnam Neyshabur, and Hanie Sedghi. Exploring the limits of large scale pre-training. *arXiv preprint arXiv:2110.02095*, 2021.

[25] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[26] Weixin Liang and James Zou. Metashift: A dataset of datasets for evaluating contextual distribution shifts and training conflicts. *arXiv preprint arXiv:2202.06523*, 2022.

[27] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[28] Michael Zhang, Nimit S Sohoni, Hongyang R Zhang, Chelsea Finn, and Christopher Ré. Correct-n-contrast: A contrastive approach for improving robustness to spurious correlations. *arXiv preprint arXiv:2203.01517*, 2022.

[29] Elan Rosenfeld, Pradeep Ravikumar, and Andrej Risteski. Domain-adjusted regression or: Erm may already learn features sufficient for out-of-distribution generalization. *arXiv preprint arXiv:2202.06856*, 2022.

[30] Aditya Krishna Menon, Ankit Singh Rawat, and Sanjiv Kumar. Overparameterisation and worst-case generalisation: friend or foe? In *International Conference on Learning Representations*, 2021.

[31] Zhao Wang and Aron Culotta. Identifying spurious correlations for robust text classification. *arXiv preprint arXiv:2010.02458*, 2020.

[32] Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Last layer re-training is sufficient for robustness to spurious correlations. *arXiv preprint arXiv:2204.02937*, 2022.

[33] Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. In *International Conference on Machine Learning*, pages 6781–6792. PMLR, 2021.

[34] Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. Learning from failure: De-biasing classifier from biased classifier. *Advances in Neural Information Processing Systems*, 33:20673–20684, 2020.

[35] John C Duchi, Tatsunori Hashimoto, and Hongseok Namkoong. Distributionally robust losses against mixture covariate shifts. *Under review*, 2, 2019.

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

[37] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[38] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[39] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020.

[40] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021.

[41] Hong Liu, Jeff Z HaoChen, Adrien Gaidon, and Tengyu Ma. Self-supervised learning is more robust to dataset imbalance. *arXiv preprint arXiv:2110.05025*, 2021.

[42] Ziyu Jiang, Tianlong Chen, Bobak Mortazavi, and Zhangyang Wang. Self-damaging contrastive learning. *arXiv preprint arXiv:2106.02990*, 2021.

[43] Ziyu Jiang, Tianlong Chen, Ting Chen, and Zhangyang Wang. Improving contrastive learning on imbalanced data via open-world sampling. *Advances in Neural Information Processing Systems*, 34, 2021.

[44] Joshua Robinson, Li Sun, Ke Yu, Kayhan Batmanghelich, Stefanie Jegelka, and Suvrit Sra. Can contrastive learning avoid shortcut solutions? *arXiv preprint arXiv:2106.11230*, 2021.

[45] Yao-Hung Hubert Tsai, Yue Wu, Ruslan Salakhutdinov, and Louis-Philippe Morency. Demystifying self-supervised learning: An information-theoretical framework. *arXiv e-prints*, pages arXiv–2006, 2020.

[46] Jiaming Song, Pratyusha Kalluri, Aditya Grover, Shengjia Zhao, and Stefano Ermon. Learning controllable fair representations. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2164–2173. PMLR, 2019.

[47] Tan Wang, Zhongqi Yue, Jianqiang Huang, Qianru Sun, and Hanwang Zhang. Self-supervised learning disentangled group representation as feature. *Advances in Neural Information Processing Systems*, 34:18225–18240, 2021.

[48] Sara Hooker, Aaron Courville, Gregory Clark, Yann Dauphin, and Andrea Frome. What do compressed deep neural networks forget? *arXiv preprint arXiv:1911.05248*, 2019.

[49] Dinghuai Zhang, Kartik Ahuja, Yilun Xu, Yisen Wang, and Aaron Courville. Can subnetwork structure be the key to out-of-distribution generalization? In *International Conference on Machine Learning*, pages 12356–12367. PMLR, 2021.

[50] Hattie Zhou, Ankit Vani, Hugo Larochelle, and Aaron Courville. Fortuitous forgetting in connectionist networks. *arXiv preprint arXiv:2202.00155*, 2022.

[51] Cian Eastwood and Christopher KI Williams. A framework for the quantitative evaluation of disentangled representations. In *International Conference on Learning Representations*, 2018.

[52] Karl Ridgeway and Michael C Mozer. Learning deep disentangled embeddings with the f-statistic loss. *Advances in neural information processing systems*, 31, 2018.

[53] Jun-Ting Hsieh, Bingbin Liu, De-An Huang, Li F Fei-Fei, and Juan Carlos Niebles. Learning to decompose and disentangle representations for video prediction. *Advances in neural information processing systems*, 31, 2018.

[54] Ruichu Cai, Zijian Li, Pengfei Wei, Jie Qiao, Kun Zhang, and Zhifeng Hao. Learning disentangled semantic representation for domain adaptation. In *IJCAI: proceedings of the conference*, volume 2019, page 2060. NIH Public Access, 2019.

[55] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016.

[56] Ricky TQ Chen, Xuechen Li, Roger Grosse, and David Duvenaud. Isolating sources of disentanglement in variational autoencoders. *arXiv preprint arXiv:1802.04942*, 2018.

[57] Yizhe Zhu, Martin Renqiang Min, Asim Kadav, and Hans Peter Graf. S3vae: Self-supervised sequential vae for representation disentanglement and data generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6538–6547, 2020.

[58] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems*, 29, 2016.

[59] Zinan Lin, Kiran Thekumparampil, Giulia Fanti, and Sewoong Oh. Infogan-cr and modelcentrality: Self-supervised model training and selection for disentangling gans. In *international conference on machine learning*, pages 6127–6139. PMLR, 2020.

[60] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *International conference on machine learning*, pages 325–333. PMLR, 2013.

[61] Harrison Edwards and Amos Storkey. Censoring representations with an adversary. *arXiv preprint arXiv:1511.05897*, 2015.

[62] Qizhe Xie, Zihang Dai, Yulun Du, Eduard Hovy, and Graham Neubig. Controllable invariance through adversarial feature learning. *arXiv preprint arXiv:1705.11122*, 2017.

[63] Alex Beutel, Jilin Chen, Zhe Zhao, and Ed H Chi. Data decisions and theoretical implications when adversarially learning fair representations. *arXiv preprint arXiv:1707.00075*, 2017.

[64] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 335–340, 2018.

[65] David Madras, Elliot Creager, Toniann Pitassi, and Richard Zemel. Learning adversarially fair and transferable representations. In *International Conference on Machine Learning*, pages 3384–3393. PMLR, 2018.

[66] Tameem Adel, Isabel Valera, Zoubin Ghahramani, and Adrian Weller. One-network adversarial fairness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 2412–2420, 2019.

[67] Han Zhao and Geoffrey J Gordon. Inherent tradeoffs in learning fair representations. *arXiv preprint arXiv:1906.08386*, 2019.

[68] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[69] Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex Smola. A kernel method for the two-sample-problem. *Advances in neural information processing systems*, 19:513–520, 2006.

[70] Christos Louizos, Kevin Swersky, Yujia Li, Max Welling, and Richard Zemel. The variational fair autoencoder. *arXiv preprint arXiv:1511.00830*, 2015.

[71] Alexander Amini, Ava P Soleimany, Wilko Schwarting, Sangeeta N Bhatia, and Daniela Rus. Uncovering and mitigating algorithmic bias through learned latent structure. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 289–295, 2019.

[72] Alexander Alemi, Ben Poole, Ian Fischer, Joshua Dillon, Rif A Saurous, and Kevin Murphy. Fixing a broken elbo. In *International Conference on Machine Learning*, pages 159–168. PMLR, 2018.

[73] Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-vae. *arXiv preprint arXiv:1804.03599*, 2018.

[74] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, pages 2649–2658. PMLR, 2018.

[75] Babak Esmaeili, Hao Wu, Sarthak Jain, Alican Bozkurt, Narayanaswamy Siddharth, Brooks Paige, Dana H Brooks, Jennifer Dy, and Jan-Willem Meent. Structured disentangled representations. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2525–2534. PMLR, 2019.

[76] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

Appendix

A Summary of Related Work

Spurious Correlations Spurious correlations arise in supervised learning models in a variety of domains, from medical imaging [4, 5] to natural language processing [11, 31]. A variety of approaches have been proposed to learn classifiers which do not make use of spurious information. Methods like GroupDRO [12] and DFR [32] require group information during training, while methods like JTT [33], LfF [34], CVaR DRO [35], CnC [28] 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 [36]. This can be done in a contrastive fashion using the InfoNCE loss [37], as in SimCLR [14] and MoCo [38], or without the need for negative samples at all, as in BYOL [39], SwAV [18], SimSiam [40], and Barlow Twins [19].

Learning under Dataset Imbalance and Shortcuts Self-supervised models have been found to be more robust to dataset imbalance [41, 42, 43, 41]. Prior work addressed shortcut learning in contrastive learning by adversarially modifying encoded features [44]. Other works in addressing group robustness or fairness in SSL, however, require group information or labels [45, 46, 47]. In the supervised setting, how subnetworks of a trained model can affect minority examples [48] or out-of-distribution generalization [49], and forgetting features via final-layer re-initialization [50], have also been studied.

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

B Evaluating Spurious Learning in Self-supervised Learning

		kNN (Target)		LR (Target)		LR (Group)
Dataset	**SSL Training**	**Average**	**Worst**	**Average**	**Worst**	**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 2: **Effect of resampling on worst group accuracy:** 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). kNN and LR represent the k-NN classifier or Logistic regression model, trained for the downstream tasks based on self-supervised representations. Re-sampling does not necessarily improve worst-group accuracy in the downstream task.

C Spurious vs. Core feature learning across layers

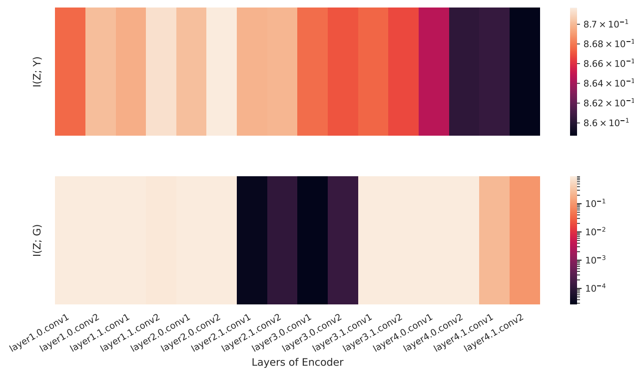

Figure 2: Comparison of Mutual Information between features and labels (top) vs. spurious attribute and labels (bottom) across layers of a ResNet-18 model trained with SimSiam on coloured MNIST. We observe that $I(Z;G)$ decreases in the intermediate layers, and grows back in the final layers, indicating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

D Experimental Setup

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 **??** we show that even an uncurated set of model transformations enhance worst-group performance), and evaluate them similar to previous experiments.

E Background

E.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:

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

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

$E = \{(x_i, y_i) \text{ s.t. } \hat{f}_{\text{id}}(x_i) \neq y_i\}$ 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.

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

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.

E.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 [14, 15, 16, 18, 19]. 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 [37] then maximizes a lower bound on the mutual information between the two views.

For instance, SimCLR [14] 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 [16] uses a siamese architecture with momentum encoders to prevent different representations from collapsing into one vector. SwAV [18] exploits online clustering for each batch to enforce consistency between cluster assignments from different views, and SimSiam [17] uses a simple stop-gradient operation in a siamese architecture to avoid collapsing.

We use SimSiam [17] 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 ${}_1 \triangleq h(f(x_1))$ and ${}_2 \triangleq f(x_2)$; SimSiam minimizes the negative cosine similarity $({}_1, {}_2)$:

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

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:

$$\mathcal{L} = \frac{1}{2}(p_1, \texttt{stopgrad}({}_2)) + \frac{1}{2}(p_2, \texttt{stopgrad}({}_1)).$$

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

E.3 Disentangled Representations

A similar line of research is creating representations where each dimension is independent and corresponds to a particular attribute [51, 52], some works study learning such representations in a supervised manner [53, 54] while unsupervised approaches rely on VAEs [55, 56, 57] and GANs [58, 59].

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 [60], using adversarial methods [61, 62, 63, 64, 65, 66, 67], or feature disentanglement based using variational approaches. [68, 69, 70, 71, 72, 73, 56, 74, 75].

Perhaps the closest to our work is [47] 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.

F Datasets

We make use of the following four image datasets:

- `waterbirds` [12]: Background (land, water) is spuriously correlated with bird type (labdbird, waterbird).
- `cmnist` (Colored MNIST): Color of digit on the images spuriously correlated with the binary class based on the number inspired by [25], with no label slipping.
- `spurcifar10` (Spurious CIFAR10) [6]: Color of lines on the images spuriously correlated with the class.
- `metashift` [26]: Cats vs Dogs task: Background (indoor, outdoor) spuriously correlated with pet type (cat, dog).

G Comparing SSL to CLIP representations

We train linear classifiers with different re-sampled sets of training examples on frozen CLIP [76] 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.

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

Table 3: Average and worst-group test accuracy of a logistic regression model trained on CLIP representations of original train, downsampled train, and upsampled train datasets. Balancing the training set used for linear evaluation helps us identify the learned representations by avoiding the statistical and geometrical skews induced by the linear evaluator.