How robust are pre-trained models to distribution shift?

Yuge Shi [*1] **Imant Daunhawer** [*2] **Julia E. Vogt** [2] **Philip H.S. Torr** [1] **Amartya Sanyal** [3]

Abstract

The vulnerability of machine learning models to spurious correlations has mostly been discussed in the context of supervised learning (SL). However, there is a lack of insight on how spurious correlations affect the performance of popular self-supervised learning (SSL) and auto-encoder based models (AE). In this work, we shed light on this by evaluating the performance of these models on both real world and synthetic distribution shift datasets. Following observations that the linear head itself can be susceptible to spurious correlations, we develop a novel evaluation scheme with the linear head trained on out-of-distribution (OOD) data, to isolate the performance of the pre-trained models from a potential bias of the linear head used for evaluation. With this new methodology, we show that SSL models are consistently more robust to distribution shifts and thus better at OOD generalisation than AE and SL models.

1. Introduction

In most real world datasets, there exist features that are irrelevant to the true labelling functions, yet highly predictive of the labels [Torralba and Efros, 2011, Calude and Longo, 2017, Fan and Lv, 2008, Fan et al., 2014]. Such features are commonly referred to as *spurious features*, while the ones that are truly relevant to labels are called *core features*. For instance, consider the Waterbirds dataset [Sagawa et al., 2020] where the classification task is to determine whether an image contains either a landbird or a waterbird. Naturally, the vast majority of the training data features landbirds on land and waterbirds on water. A classifier can therefore make a decision using only the background of the image, which is only spuriously correlated to the labels (i.e., the class of the bird), and disregard anatomical features of the bird [Kirichenko et al., 2022]. However, using just the background for classification can result in poor OOD generalisation performance on test data where the spurious correlation is no longer present, e.g., for images of a landbird flying over a river [Brendel and Bethge, 2019a, Shah et al., 2020, Geirhos et al., 2020, Singla and Feizi, 2021].

Recent work discovered that supervised learning (SL) can be very susceptible to spurious correlation [Shah et al., 2020, Geirhos et al., 2020]. This is not at all surprising—after all, the objective of SL is to most accurately predict the targets, not to represent the data most accurately. If a model's only goal is to predict the labels, there is no guarantee that it should favour core features over spurious ones; when both are equally predictive of the labels, the choice comes down to the inductive bias of the optimiser and the model architecture.

Can we get around this by replacing explicit supervision by another objective? In this work, we turn our attention to unsupervised pre-training methods, including self-supervised learning (SSL) algorithms (e.g., [Chen et al., 2020, Grill et al., 2020, Chen and He, 2021]), which learn representations by enforcing invariance between the representations of two distinctly augmented views of the same image, and auto-encoder based models (AE) [Rumelhart et al., 1985, Kingma and Welling, 2014, Higgins et al., 2017, Burda et al., 2016], which learn by reconstructing the input image. These methods have been under-explored in the context of spurious correlations, and since SSL and AE methods do not require annotations of the data, we hypothesize that they stand a better chance at avoiding the exploitation of spurious correlation and thus faring better under distribution shift compared to SL.

To understand this, we develop a suite of experiments to evaluate the performance of pre-trained models under distribution shift. Figure 1 provides a summary of our results, comparing a wide range of models from the following classes of methods: (i) SSL, (ii) AE, and (iii) SL. To the best of our knowledge, we are the first to systematically evaluate the performance of these pre-trained models under distribution shift. In addition, to isolate the performance of the pre-trained models from the potential bias of the linear layer used for prediction (as discovered in Kirichenko et al. [2022] and Kang et al. [2020]), we establish a new

[1] Department of Engineering Science, University of Oxford [2] Department of Computer Science, ETH Zurich [3] ETH AI Center, ETH Zurich. Correspondence to: Yuge Shi <yshi@robots.ox.ac.uk>.

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

evaluation scheme where the linear head is trained on OOD data (right subplots in Figures 1a to 1c). We also provide results using the classical evaluation scheme, where the linear head is trained on in-distribution (ID) data (left subplots in Figures 1a to 1c). The main takeaways from this paper are:

- ***SSL* $>$ *AE* $\geq$ *SL for distribution shift:*** Results of our proposed OOD linear head evaluation show that SSL models consistently achieve better performance than AE, however, AE models still perform at least as good as SL on both synthetic and realistic settings that we study;
- ***Performance of all models under both synthetic and realistic distribution shift can be significantly improved by retraining the last linear layer:*** We show a large performance gain evaluating the models using linear head trained on a small amount of OOD data, in contrast to the baseline where the linear head is trained on ID data (compare left to right subplots in Figure 1); the surprising gain of this cheap procedure indicates that 1) existing pre-trained models, especially SSL, avoid the exploitation of spurious correlations and 2) more work on improving the performance under distribution shift should focus on balancing the final linear layer.

2. How to evaluate spurious correlation?

Inspired by prior work, we examine the models' robustness to spurious correlation under two different settings:

Synthetic distribution shift (Section 3.1): Shah et al. [2020] studies the susceptibility of SL to spurious correlations by careful design of synthetic datasets, which contain one simple feature (e.g. color) and one complex feature (e.g. shape) that are both equally predictive of the target. They show through experiments on these datasets that SL models can completely ignore the more complex feature and predict solely based on the simple feature. This bias can be very harmful for OOD generalisation if the simple features are not present in the OOD data, or if their correlation with the true labels changes. They name this intriguing phenomenon *simplicity bias*, and their set of experiments constitute a useful set of tools for diagnosing a model's vulnerability to spurious correlation under controlled settings.

Realistic distribution shift (Section 3.2): Koh et al. [2021], Sagawa et al. [2021] propose WILDS, a benchmark for OOD generalisation in the wild. The benchmark studies *domain generalisation*, where the training data is sampled from one or many source domains, while the test data is sampled from an unseen target domain. A model's performance on the target domain is a reliable indicator of its ability to learn domain-invariant features that are predictive of the targets while ignoring spurious features that are domain-specific. One can view this as a more realistic scenario of the *simplicity bias* datasets mentioned above. However, it is not possible to disentangle the *simple spurious feature* and the *complex core feature*.

2.1. Disentangling linear head from pre-trained models

Kang et al. [2020] and Kirichenko et al. [2022] observe the interesting phenomenon that the linear classifier itself could be susceptible to a simplicity bias. They show that by simply retraining the linear classifier on data where the spurious correlation is removed, a model's performance can be drastically improved on the OOD test set. This indicates that the representations learned by the backbone are able to capture *both* the simple and the complex features in the data, however the linear classifier only assigns weights to those simpler features that are spuriously correlated to the labels. Since our interest lies in examining the vulnerability of the representations of the pre-trained models to spurious correlation, without the bias of the linear classifier, for all our experiments we report the performance using a linear head trained on the OOD data. We also provide results using more standard evaluation scheme for distribution shift, where the linear head is trained on ID data. Note that in both scenarios, the model is evaluated on the OOD test set. A summary of the results using this methodology is presented in Figure 1. The evaluation with the OOD linear head provides insight on whether the representations produced by the pre-trained models contain information about all features, and the difference between the ID and OOD linear head sheds light on the bias of the linear classifier.

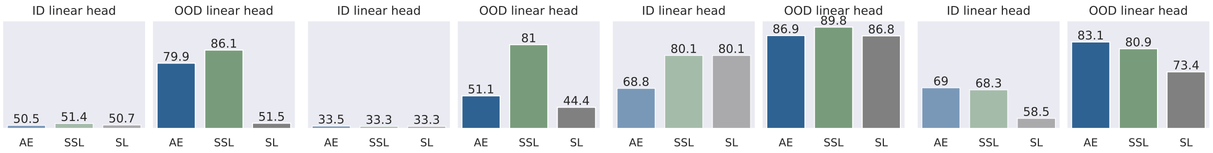
3. Experimental results

Learning algorithms: 8 in total. For all experiments we report results on eight types of learning algorithms, including three SSL algorithms 1) SimCLR [Chen et al., 2020], 2) SimSiam [Chen and He, 2021], 3) BYOL [Grill et al., 2020]; four AE algorithms 1) Autoencoder [Rumelhart et al., 1985], 2) Variational Autoencoder (VAE) [Kingma and Welling, 2014], 3) β-VAE [Higgins et al., 2017] and 4) Importance Weighted Autoencoder (IWAE) [Burda et al., 2016]; as well as one model trained using supervised learning (SL). We perform a hyperparameter search on learning rate, scheduler, optimiser, representation size, etc. for each model, and we use the same set of augmentations for all models to ensure a fair comparison. See Appendix C for more details.

3.1. Synthetic distribution shift

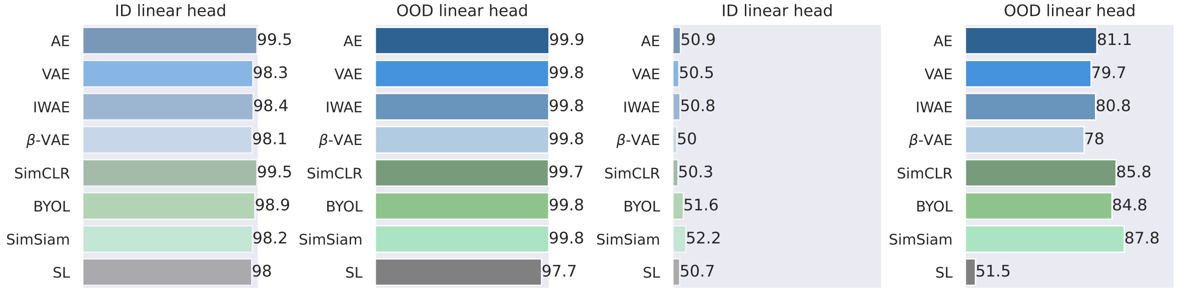
Findings: *With a perfect correlation between the complex and simple features in the data, representations learned by SSL and AE models capture both features, while SL focusses on the simpler features only.*

We are interested in understanding whether models pre-trained with SL, SSL and AE methods are vulnerable to



(a) MNIST-CIFAR (b) CdSprites (c) Camelyon17 (d) Camelyon17_SB

Figure 1: OOD performance of auto-encoder based (AE), self-supervised learning (SSL), supervised learning (SL) models. For each dataset, we evaluate on the OOD test set with linear heads trained on ID (left, transparent) and OOD data (right, solid) respectively. In each subplot, the lowest visible value corresponds to the relative frequency of the majority class from the respective dataset.



(a) MNIST, OOD acc (%). (b) CIFAR, OOD acc (%).

Figure 2: Evaluations on MNIST-CIFAR dataset. Blue colors are AE models, green colors are SSL models and grey is SL. **Fig (a), (b), left:** Accuracy on OOD MNIST and CIFAR with linear head trained on ID data; **Fig (a), (b), right:** Accuracy on OOD MNIST and CIFAR with linear head trained on OOD data.

simplicity bias under synthetic distribution shift settings. Following Shah et al. [2020], Shi et al. [2021], we evaluate this using the MNIST-CIFAR and CdSprites datasets.

3.1.1. MNIST-CIFAR

Following Shah et al. [2020], we use the MNIST-CIFAR dataset to examine the simplicity bias of different models. The dataset consists of concatenations of images from 2 classes of MNIST and CIFAR-10. It contains the following splits with varying correlation between the labels of the MNIST and CIFAR-10 images:

- **ID train**: 1.0 correlation between MNIST and CIFAR-10 labels. Contains two classes: class 0 with MNIST “0” and CIFAR-10 “automobile”, and class 1 with MNIST “1” and CIFAR-10 “plane” (Figure 5a);
- **OOD train**: no correlation between MNIST and CIFAR-10 labels, images from the two classes of MNIST and CIFAR-10 are randomly paired (Figure 5b);
- **OOD test**: generated similarly to the OOD train set using the test set of MNIST and CIFAR-10 (Figure 5b).

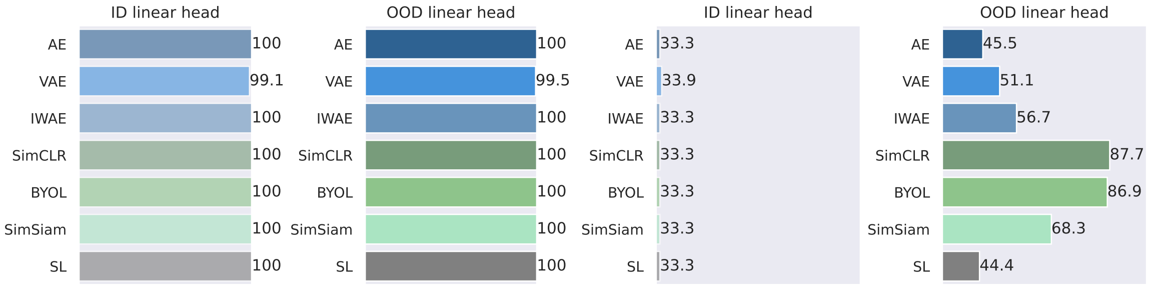
We train a CNN backbone on the ID train set using the respective SL, SSL, or AE algorithm. At test time, we freeze the backbone and train two linear heads, one on ID train and the other on OOD train, and evaluate their performance on the OOD test set. This helps us disentangle the simplicity bias of the representation from that of the linear head.

In Figures 1a and 2, we observe that with ID linear head, all models exhibit a simplicity bias, as the classifiers reach near 100% accuracy for the classification of the MNIST digit (Figure 2a, left) but near random accuracy for predicting the CIFAR class (Figure 2b, left). However, the OOD linear head results indicate that both SSL and AE models are able to learn representations that capture *both* features, with not only near perfect accuracy on MNIST (Figure 2a, right), but also $\sim 80\%$ accuracy on CIFAR-10 (Figure 2b, right). On the other hand, representations learned using SL do not extract (linearly separable) CIFAR-10 features, since its OOD evaluation performance is near random on CIFAR-10.

Interestingly, Figure 1a shows that using OOD linear head, the average performance of models trained using SSL is ~ 5 percentage points higher than that of AE, while AE scores ~ 30 percentage points higher than SL.

3.1.2. CdSprites

CdSprites [Shi et al., 2021] is a colored variant of the popular dSprites dataset [Matthey et al., 2017], which consists of images of 2D shapes that are procedurally generated from 6 ground truth factors (see Figure 6 for examples). To induce a spurious correlation between the color and shape features, the sprites are colored conditionally on the shape and depending on the correlation coefficient ρ that is set when generating the dataset. We generate five versions of the CdSprites dataset with a different correlation



(a) Color classification, OOD acc (%).

(b) Shape classification, OOD acc (%).

Figure 3: Evaluations on the CdSprites dataset. Blue colors are AE models, green colors are SSL models and grey is SL. **Figure 3a:** Color classification accuracy on the test set with a linear head trained on ID data (left, transparent) and OOD data (right, solid); **Figure 3b:** Shape classification accuracy on the OOD test set with a linear head trained on ID data (left, transparent) and OOD data (right, solid).

coefficient $\rho \in \{0, 0.25, 0.5, 0.75, 1.0\}$ respectively and use the following splits:

- **ID train**: CdSprites dataset with correlation ρ_{id} between shape and color features;
- **OOD train**: CdSprites dataset with uncorrelated shape and color features, i.e., $\rho_{\text{ood}} = 0$;
- **OOD test**: generated similarly to the OOD train set.

Based on the above splits, we use the same evaluation protocol as for MNIST-CIFAR (see Section 3.1.1) to disentangle the simplicity bias of the representation from the simplicity bias of the linear classification head.

Figures 1b and 3 show the results for backbones trained on data with a perfect correlation ($\rho_{\text{id}} = 1$) between shape and color features. Similar to MNIST-CIFAR, with the ID linear head we observe a simplicity bias, where the performance on the simple feature (color, Figure 3a) is perfect and the performance on the complex feature (shape, Figure 3b) is trivial. Again, the OOD linear head results (right subplots of Figures 3a and 3b respectively) indicate that AE and especially SSL models are able to learn representations that capture *both* shape and color features. We also provide results with changing values of ρ_{id} and ρ_{ood} in Appendix B.1.

3.2. Real-world distribution shift

Findings: *Learning with labels (SL) is no better than learning without labels (SSL, AE) for OOD generalisation on real-world distribution shift problems, and the performance of all models can be significantly improved by retraining the linear head on a small amount of OOD data.*

In this section we investigate the performance of different pre-trained models on real-world distribution shift tasks. We utilise the Camelyon17 dataset from the WILDS benchmark [Koh et al., 2021], which contains scans of lymph node sections acquired from different hospitals, and the task is to determine whether a given patch contains breast cancer tissue. In addition to evaluating on the original dataset, we create a simplicity bias (SB) version of Camelyon17 (Camelyon17_SB) by sub-sampling the ID train split of the Camelyon17 dataset such that the label and domain of each sample are perfectly correlated. This allows us to examine the OOD generalisation performance of different models on a more challenging but still realistic distribution shift task.

3.3. Camelyon17

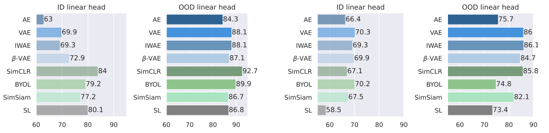
The original Camelyon17 dataset from WILDS benchmark contains three splits using tissue scans from five hospitals: train (3 hospitals), validation (1 hospital) and test (1 hospital). We further create four splits specified as follows:

- **ID train**: Same as the train set of Camelyon17;
- **OOD val.**: Same as the validation set of Camelyon17;
- **OOD train**: Contains 10% of the data from the test set;
- **OOD test**: Contains the held-out 90% of test data.

Similar to our previous experiments, we train the linear heads on ID train and OOD train respectively, and evaluate all models on the OOD test set. We use 10-fold cross-validation for the OOD train and test set. Following WILDS, we adopt DenseNet-121 [Huang et al., 2017] as backbone for all models, and the OOD validation set is used to perform early stopping and hyperparameter search. The main results are shown in Figures 1c and 4a and we provide more detailed numerical results with standard deviation values in Table 1.

With the linear head trained on ID data, the left subplots in Figures 1c and 4a show that representations learned using SSL algorithms result in similar OOD test accuracy as those trained using SL, while the OOD accuracy from AE models is $\sim$10 percentage points lower.

However, the OOD test accuracy of AE algorithms catches up to those from SL and SSL algorithms when we use the



(a) Camelyon17, OOD test acc. (%)

(b) Camelyon17_SB, OOD test acc. (%)

Figure 4: Evaluations on the two variations of Camelyon17 dataset. Blue colors are AE models, green colors are SSL models and grey is SL. **Left:** Accuracy on OOD test set with the linear head trained on ID data; **Right:** Accuracy on OOD test set with the linear head trained on OOD data.

OOD linear head for evaluation. Figures 1c and 4a (right subplots) show that SSL models achieve the best performance, out-performing AE and SL by nearly 3 percentage points. This demonstrates that the representations learned by SSL models are more robust against distribution shift than AE and SL models, which highlights a surprising adverse effect of labels for OOD generalisation in this setting. Another significant finding of this experiment is that by using only 10% of the data from the original test split to train the linear classification head, we are able to improve the performance of all models by 13 percentage points on average. All of this shows that the bias of the linear classification head plays a significant role even for real world distribution shifts, and that it is possible to mitigate this effect by training the linear head using a small amount of OOD data.

3.4. Camelyon17_SB

To further examine models' ability to generalise OOD, we adjust the subpopulation of the training split of the Camelyon17 dataset to create a *simplicity bias* version of this realistic dataset. Specifically, the ID train split of Camelyon17 contains tissue scans from 3 hospitals, that are labelled as either "benign" or "malignant". To construct the train set of Camelyon17_SB, we take only the "benign" samples from hospital 1 and 2, and only the "malignant" samples from hospital 3. As such, the label and domain of examples in the train set are spuriously correlated. Given the texture bias of CNNs [Geirhos et al., 2019, Brendel and Bethge, 2019b], the model is prone to trivially minimising the training loss by using only domain information for classification, which would result in poor OOD generalisation performance. Using the same experimental setup as in Section 3.3, we report the results for Camelyon17_SB in Figures 1d and 4b.

Comparing the performance of models on the biased version of Camelyon17 (Figures 1d and 4b) to the performance on the original dataset (Figures 1c and 4a), we see that surprisingly, AE models experience almost no performance drop for both the ID and OOD linear head evaluation; SL model, on the other hand, sees the most significant decline in its OOD generalisation performance, showing a 22% accuracy decrease with ID linear head and 13% with OOD linear head. Additionally, consistent to our previous findings, unsupervised models (AE, SSL) outperforms supervised model (SL) on this dataset. We also note that while with the original Camelyon17 dataset, SL model's performance catches up with the rest when using OOD linear head for evaluation, on Camelyon17_SB there still exist a 10% performance gap between SL and other unsupervised models when switching to OOD linear head. The results further demonstrate that unsupervised models including AE and SSL are better at OOD generalisation than SL models, and their advantages become even more significant under this more challenging distribution shift setting.

4. Conclusion and future work

In this paper, we investigate the performance of both unsupervised (AE, SSL) and supervised (SL) pre-training methods for OOD generalisation. Through extensive and principled experiments on both synthetic and real-world distribution shift tasks, we find unsupervised models to consistently outperform supervised models. This is surprising, since most work in domain generalisation and domain adaptation is based on supervised learning; yet, our findings clearly suggest that unsupervised learning methods can be far superior for these tasks. We plan to extend this work by evaluating AE, SSL and SL models on more datasets from the WILDS benchmark.

Another key contribution of this work is our OOD linear head evaluation scheme, which helps us isolate the performance of the pre-trained models from the potential bias of the final linear head. We claim that this change is necessary to be able to make comparisons between various pretraining methods irreespective of the final downstream task. We find the OOD generalisation performance of all models to be significantly improved by re-training the linear head on small amount of OOD data. In future work, we will investigate 1) the performance of a nonlinear prediction head trained on ID and OOD data respectively and 2) the sample efficiency (i.e., the amount of OOD data needed) for training the linear and nonlinear prediction heads.

Acknowledgements

YS and PHST were supported by the UKRI grant: Turing AI Fellowship EP/W002981/1 and EPSRC/MURI grant: EP/N019474/1. We would also like to thank the Royal Academy of Engineering and FiveAI. YS was additionally supported by Remarkdip through their PhD Scholarship Programme. ID was supported by the SNSF grant #200021_188466. AS was partially supported by the ETH AI Center postdoctoral fellowship. Special thanks to Alain Ryser for suggesting the design of a biased version of the Camelyon17 dataset.

References

Wieland Brendel and Matthias Bethge. Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019a. URL https://openreview.net/forum?id=SkfMWhAqYQ.

Wieland Brendel and Matthias Bethge. Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019b. URL https://openreview.net/forum?id=SkfMWhAqYQ.

Yuri Burda, Roger B. Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL http://arxiv.org/abs/1509.00519.

Cristian S Calude and Giuseppe Longo. The deluge of spurious correlations in big data. *Foundations of science*, 22(3):595–612, 2017.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 2020. URL http://proceedings.mlr.press/v119/chen20j.html.

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.

Victor Guilherme Turrisi da Costa, Enrico Fini, Moin Nabi, Nicu Sebe, and Elisa Ricci. solo-learn: A library of self-supervised methods for visual representation learning. *Journal of Machine Learning Research*, 23(56):1–6, 2022. URL http://jmlr.org/papers/v23/21-1155.html.

Jianqing Fan and Jinchi Lv. Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70 (5):849–911, 2008.

Jianqing Fan, Fang Han, and Han Liu. Challenges of big data analysis. *National science review*, 1(2):293–314, 2014.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=Bygh9j09KX.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent - A new approach to self-supervised learning. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/f3ada80d5c4ee70142b17b8192b2958e-Abstract.html.

Irina Higgins, Loïc 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. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=Sy2fzU9gl.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 2261–2269. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.243. URL https://doi.org/10.1109/CVPR.2017.243.

Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=r1gRTCVFvB.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6114.

Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Last layer re-training is sufficient for robustness to spurious correlations. *ArXiv preprint*, abs/2204.02937, 2022. URL https://arxiv.org/abs/2204.02937.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton Earnshaw, Imran Haque, Sara M. Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. WILDS: A benchmark of in-the-wild distribution shifts. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 5637–5664. PMLR, 2021. URL http://proceedings.mlr.press/v139/koh21a.html.

Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=ryxGuJrFvS.

Shiori Sagawa, Pang Wei Koh, Tony Lee, Irena Gao, Sang Michael Xie, Kendrick Shen, Ananya Kumar, Weihua Hu, Michihiro Yasunaga, Henrik Marklund, et al. Extending the wilds benchmark for unsupervised adaptation. *ArXiv preprint*, abs/2112.05090, 2021. URL https://arxiv.org/abs/2112.05090.

Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/6cfe0e6127fa25df2a0ef2ae1067d915-Abstract.html.

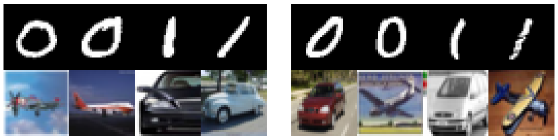
Yuge Shi, Jeffrey Seely, Philip HS Torr, N Siddharth, Awni Hannun, Nicolas Usunier, and Gabriel Synnaeve. Gradient matching for domain generalization. *ArXiv preprint*, abs/2104.09937, 2021. URL https://arxiv.org/abs/2104.09937.

Sahil Singla and Soheil Feizi. Salient imagenet: How to discover spurious features in deep learning? In *International Conference on Learning Representations*, 2021.

Antonio Torralba and Alexei A. Efros. Unbiased look at dataset bias. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*, pages 1521–1528. IEEE Computer Society, 2011. doi: 10.1109/CVPR.2011.5995347. URL https://doi.org/10.1109/CVPR.2011.5995347.

A. Dataset visualisation

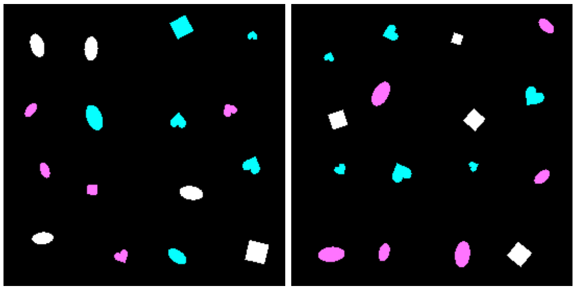
In Figure 5 and **??** we visualise the ID and OOD split of MNIST-CIFAR and CdSprites dataset.



(a) ID split, fixed matching with `0-plane`, `1-car`.

(b) OOD split, random matching of classes.

Figure 5: MNIST-CIFAR dataset.



(a) $\rho = 0$

(b) $\rho = 1$

Figure 6: CdSprites dataset. Each subplot shows 16 samples from the dataset generated with a given correlation ρ between shape and color features.

B. Additional Experimental Results

B.1. CdSprites

In addition to our results in Figure 3, where we use a dataset with perfectly correlated features ($\rho_{\text{id}} = 1$) to train the backbones, in Figure 7 we vary ρ_{id} to analyse the effect of imperfectly correlated features. Notably, with imperfect correlation ($\rho_{\text{id}} < 1$), the OOD linear heads trained on top of the SL and SSL backbones perform perfectly. For the AE, we observe that the performance of the OOD linear head does not depend on the correlation ρ_{id} in the data used to train the backbones. Our results suggest that with imperfect correlation between features, SL and SSL models learn a linearly separable representation of the features, whereas AE does not.

In Figure 8 we provide an ablation where we also vary ρ_{ood}, the correlation in the data used to train and evaluate the linear head. Figures 8b and 8c corroborate our results that SSL performs on par with SL for $\rho_{\text{id}} < 1$ and strictly better when $\rho_{\text{id}} = 1$. For the AE (Figure 8a), we observe an interesting pattern where the performance of the OOD linear head depends on the OOD correlation ρ_{ood}, but not on the correlation ρ_{id} in the data used to train the backbones. Hence, the ablation corroborates our result that SL and SSL models learn a linearly separable representation of the shape and color features when there is an imperfect correlation between the features, whereas AE does not.

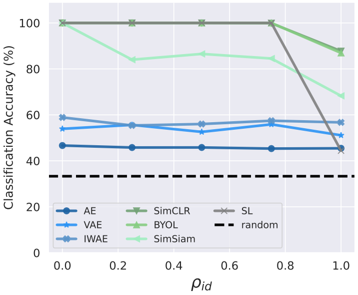
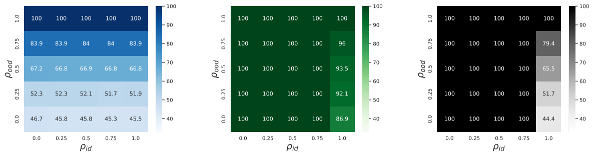


Figure 7: Evaluations on CdSprites with varying correlation ρ_{id}. Shape classification accuracy as a function of the correlation between shape and color features (ρ_{id}, x-axis) in the ID train split used to pre-train the respective backbone. Linear heads were trained on top of the frozen backbones using the OOD train split. Blue colors are AE models, green colors are SSL models, and grey is SL. The black horizontal line denotes the random baseline (33.3% for three classes).



(a) AE

(b) SSL

(c) SL

Figure 8: Correlation coefficient ablation for CdSprites. Shape classification accuracy for the CdSprites experiment with varying correlation of the ID training data (ρ_{id}, x-axis) and OOD training and test data (ρ_{ood}, y-axis). Backbones were trained on data with correlation ρ_{id} and linear classifiers trained and evaluated on top of the frozen backbones with correlation ρ_{ood}. For the class of AE models we only report the performance of the autoencoder, whereas for the class SSL models we report the performance of BYOL, since we found no significant differences between models within each of these classes.

Table 1: Results on Camelyon17.

	ID linear head		OOD linear head	
	Val. acc.	**Test acc.**	Val. acc.	**Test acc.**
AE	65.9 (±*9.4*)	63.0 (±*3.9*)	84.2 (±*2.3*)	84.3 (±*2.3*)
VAE	69.3 (±*7.7*)	69.9 (±*3.6*)	88.1 (±*2.1*)	88.1 (±*2.1*)
IWAE	67.6 (±*6.9*)	69.3 (±*7.9*)	88.0 (±*0.9*)	88.1 (±*0.9*)
β-VAE	69.2 (±*3.2*)	72.9 (±*7.9*)	87.0 (±*4.4*)	87.1 (±*4.4*)
SimCLR	81.0 (±*4.5*)	84.0 (±*3.4*)	92.7 (±*1.7*)	92.7 (±*1.7*)
BYOL	83.1 (±*2.5*)	79.2 (±*4.1*)	89.8 (±*2.0*)	89.9 (±*2.0*)
SimSiam	74.0 (±*5.6*)	77.2 (±*6.9*)	86.6 (±*1.5*)	86.7 (±*1.5*)
SL	79.2 (±*2.3*)	80.1 (±*3.1*)	86.8 (±*2.1*)	86.8 (±*2.1*)
WILDS, ERM	84.9 (±*3.1*)	70.8 (±*7.2*)	-	-

B.2. Camelyon17

Following [Koh et al., 2021] we report results averaged over 10 random seeds. Please refer to Table 1 for numerical results with standard deviation values. Note that our SL model uses the same training objective as the empirical risk minimisation (ERM) algorithm reported on the original WILDS benchmark, however the results are quite different. We believe that our performance gain on the test set is resulted from the image augmentations that we use for all models to ensure fair comparison to the augmentation-based SSL models.

C. Architecture and Hyperparameters

In this appendix we list the architecture and hyperparameters used in our experiments. Our code is developed on the amazing `solo-learn` code base [da Costa et al., 2022], which is originally developed as a library for SSL algorithms. Please see implementation details for each dataset in the respective subsection.

C.1. MNIST-CIFAR

We use the same hyperparameter search range for models in each category of AE, SSL and SL, as outlined in Table 2. The chosen hyperparameters for each model are specified in Table 3.

In Shah et al. [2020] where MNIST-CIFAR was originally proposed, authors utilised more complex backbone architecture such as DenseNet and MobileNet. However in our experiments, we find that a lightweight 4-layer CNN can already achieve very high accuracy on both MNIST and CIFAR (see Figures 2a and 2b, right). The architecture of the CNN we use can be found in Table 4. Note that for SL and SSL we only use the encoder and for AE we use the decoder as well. The size of base channel C and latent dimension L are found through hyperparameter search.

Table 3: Chosen hyperparameters for MNIST-CIFAR including latent dimension (L), base feature size of CNN (C), batch size (B), learning rate (*lr.*), weight decay (*wd.*), optimiser (*optim.*), learning rate scheduler (*lr.scheduler*).

	L	C	B	lr.	wd.	Optim.	lr. scheduler
AE	128	16	128	1e-3	0	Adam	warmup cosine
VAE	128	32	128	1e-4	0	Adam	warmup cosine
IWAE	128	32	128	1e-4	0	Adam	step
β-VAE	128	16	128	1e-4	0	Adam	step
SimCLR	128	32	128	6e-1	1e-4	SGD	warmup cosine
BYOL	128	64	128	7e-1	0	SGD	warmup cosine
SimSiam	128	128	128	6e-1	1e-5	SGD	warmup cosine
Supervised	128	16	128	1e-4	0	SGD	warmup cosine

Encoder
Input $\in \mathbb{R}^{3\times 64\times 32}$
4x4 conv. C stride 2x2 pad 1x1 & ReLU
4x4 conv. 2C stride 2x2 pad 1x1 & ReLU
4x4 conv. 4C stride 2x2 pad 1x1 & ReLU
4x1 conv. 4C stride 2x1 pad 1x0 & ReLU
4x4 conv. L stride 1 pad 0, 4x4 conv. L stride 1x1 pad 0x0
Decoder
Input $\in \mathbb{R}^{L}$
4x4 upconv. 4C stride 1x1 pad 0x0 & ReLU
4x1 upconv. 4C stride 2x1 pad 1x0 & ReLU
4x4 upconv. 2C stride 2x2 pad 1x1 & ReLU
4x4 upconv. C stride 2x2 pad 1x1 & ReLU
4x4 upconv. 3 stride 2x2 pad 1x1 & Sigmoid

Table 4: CNN architecture, MNIST-CIFAR dataset.

C.2. CdSprites

We found all models to be relatively robust to hyperparameters, as most configurations result in close to perfect shape and color classification accuracy on the ID validation set. The chosen hyperparameters for each model are specified in Table 5. We omit β-VAE from the comparison, as we empirically found that $\beta = 1$ leads to the best performance on the ID validation set and therefore the results for the β-VAE would be similar to the VAE. We use the same augmentations (random crops and horizontal flips) for all models and use no color augmentations in order to keep the invariance of the learned representations with respect to color. The encoder and decoder architectures are described in Table 6.

Table 2: Hyperparameter search range for MNIST-CIFAR, including base channel size of CNN (C), learning rate (*lr.*), weight decay (*wd.*), optimiser (*optim.*), learning rate scheduler (*lr. scheduler*).

	C	lr.	wd.	Optim.	lr. scheduler
AE	{16, 32, 64, 128}	{1e-4, 5e-4, 1e-3, 5e-3, 1e-2}	{0, 1e-4}	{Adam, SGD}	{warmup cosine, step, none}
SSL	{16, 32, 64, 128}	uniformly sampled from [0.1, 1]	{0, 1e-4}	{Adam, SGD}	{warmup cosine, step, none}
SL	{16, 32, 64, 128}	{1e-4, 5e-4, 1e-3, 5e-3, 1e-2, 1e-1, 5e-1}	{0, 1e-4}	{Adam, SGD}	{warmup cosine, step, none}

Table 5: Chosen hyperparameters for CdSprites including latent dimension (L), base feature size of CNN (C), batch size (B), learning rate (*lr.*), weight decay (*wd.*), optimiser (*optim.*), learning rate scheduler (*lr.scheduler*).

	L	C	B	lr.	wd.	Optim.	none
AE	512	64	128	5e-5	1e-4	Adam	none
VAE	512	64	128	5e-5	1e-4	Adam	none
IWAE	512	64	128	5e-5	1e-4	Adam	none
SimCLR	64	32	64	5e-3	1e-5	SGD	warmup cosine
BYOL	64	32	64	5e-1	1e-5	SGD	warmup cosine
SimSiam	64	32	64	8e-2	1e-5	SGD	warmup cosine
Supervised	512	64	128	5e-5	1e-4	Adam	none

Encoder
Input $\in \mathbb{R}^{3\times 64\times 64}$
4x4 conv. C stride 2x2 pad 1x1 & ReLU
4x4 conv. $2C$ stride 2x2 pad 1x1 & ReLU
4x4 conv. $4C$ stride 2x2 pad 1x1 & ReLU
4x4 conv. $8C$ stride 2x2 pad 1x1 & ReLU
4x4 conv. L stride 1 pad 0
Decoder
Input $\in \mathbb{R}^{L}$
4x4 upconv. $8C$ stride 1x1 pad 0x0 & ReLU
4x4 upconv. $4C$ stride 2x2 pad 1x1 & ReLU
4x4 upconv. $2C$ stride 2x2 pad 1x1 & ReLU
4x4 upconv. C stride 2x2 pad 1x1 & ReLU
4x4 upconv. 3 stride 2x2 pad 1x1

Table 6: CNN architecture, CdSprites dataset.

C.3. Camelyon17

For hyperparameters including batch size, max epoch and model selection criteria, we follow the same protocol as in WILDS [Koh et al., 2021] and use a batch size of 32, train all models for 10 epochs and select the model that results in the highest accuracy on the validation set. For the rest, we use the same hyperparameter search range for models in each category of AE, SSL and SL, as outlined in Table 7. The chosen hyperparameters for each model are specified in Table 8.

Table 8: Chosen hyperparameters for Camelyon17 including latent dimension (L), learning rate (*lr.*), weight decay (*wd.*), optimiser (*optim.*), learning rate scheduler (*lr.scheduler*).

	Decoder	lr.	wd.	Optim.	lr. scheduler
AE	ResNet	5e-4	1e-5	SGD	warmup cosine
VAE	MLP	1e-4	0	Adam	none
IWAE	MLP	1e-4	0	Adam	none
β-VAE	MLP	1e-4	0	Adam	none
SimCLR	-	1e-1	0	SGD	none
BYOL	-	1e-1	1e-5	SGD	warmup cosine
SimSiam	-	1e-1	1e-5	SGD	warmup cosine
Supervised	-	1e-3	1e-3	SGD	none

We follow Koh et al. [2021] and use DenseNet121 [Huang et al., 2017] as backbone architecture. For the decoder of the AE models, we perform hyperparameter search between three architectures: a CNN (see Table 9), a simple 3-layer MLP (see Table 10) and a ResNet-like decoder with skip connections (see Table 11).

CNN, Decoder
Input $\in \mathbb{R}^{L}$
4x4 upconv. $8C$ stride 2x2 pad 1x1 & ReLU
4x4 upconv. $8C$ stride 2x2 pad 0x0 & ReLU
4x4 upconv. $4C$ stride 2x2 pad 1x1 & ReLU
4x4 upconv. $2C$ stride 2x2 pad 1x1 & ReLU
4x4 upconv. C stride 2x2 pad 1x1 & ReLU
4x4 upconv. 3 stride 2x2 pad 1x1 & Sigmoid

Table 9: CNN architecture, Camelyon17 dataset.

MLP, Decoder
Input $\in \mathbb{R}^{L}$
fc. $2L$ & ReLU
fc. $4L$ & ReLU
fc. 3*96*96 & ReLU

Table 10: MLP architecture, Camelyon17 dataset.

Table 7: Hyperparameter search range for Camelyon17, including decoder type, latent dimension (L), learning rate (*lr.*), weight decay (*wd.*), optimiser (*optim.*), learning rate scheduler (*lr. scheduler*).

	Decoder type	L	lr.	wd.	Optim.	lr. scheduler
AE	[CNN, MLP, ResNet]	{256, 512, 1024}	{1e-4, 5e-4, 1e-3, 5e-3, 1e-2}	{0, 1e-4}	{Adam, SGD}	{warmup cosine, step, none}
SSL	-	{256, 512, 1024}	{1e-4, 5e-4, 1e-3, 5e-3, 1e-2, 1e-1, 5e-1, 1}	{0, 1e-3, 1e-4, 1e-5}	{Adam, SGD}	{warmup cosine, step, none}
SL	-	{256, 512, 1024}	{1e-4, 5e-4, 1e-3, 5e-3, 1e-2, 1e-1, 5e-1}	{0, 1e-4}	{Adam, SGD}	{warmup cosine, step, none}

ResNet, Decoder
Input $\in \mathbb{R}^L$
fc. 2048 & ReLU
3x3 conv. 16C stride 1x1 pad 1x1
3x3 conv. 16C stride 1x1 pad 1x1
x2 upsample
3x3 conv. 8C stride 1x1 pad 1x1
3x3 conv. 8C stride 1x1 pad 1x1
x2 upsample
3x3 conv. 8C stride 1x1 pad 1x1
3x3 conv. 8C stride 1x1 pad 1x1
3x3 conv. 3 stride 1x1 pad 1x1

Table 11: ResNet decoder architecture, Camelyon17 dataset.