Identifying and Disentangling Spurious Features in Pretrained Image Representations

Rafayel Darbinyan [1 2] Hrayr Harutyunyan [3] Aram H. Markosyan [4] Hrant Khachatrian [1 2]

Abstract

Neural networks employ spurious correlations in their predictions, resulting in decreased performance when these correlations do not hold. Recent works suggest fixing pretrained representations and training a classification head that does not use spurious features. We investigate how spurious features are represented in pretrained representations and explore strategies for removing information about spurious features. Considering the `Waterbirds` dataset and a few pretrained representations, we find that even with full knowledge of spurious features, their removal is not straightforward due to entangled representation. To address this, we propose a linear autoencoder training method to separate the representation into core, spurious, and other features. We propose two effective spurious feature removal approaches that are applied to the encoding and significantly improve classification performance measured by worst group accuracy.

1. Introduction

In many classification datasets, some features are predictive of the label but are not causally related. It is often said that these features are *spuriously correlated* with the label, as their correlation might not hold for data collected in another environment. For example, suppose we collect typical images of cows and camels and form a binary classification task. In that case, we will find that the background is correlated with the label, as cows are often photographed in barns or green pastures, while camels are often photographed in deserts (Beery et al., 2018). However, this correlation will be *spurious* as the background information is not causally related to the label, and we can easily make another dataset of cows and camels in which this correlation does not hold.

It is well-established that neural networks are susceptible to spurious correlations (Torralba & Efros, 2011; Ribeiro et al., 2016; Gururangan et al., 2018; Zech et al., 2018; McCoy et al., 2019; Geirhos et al., 2019; 2020; Xiao et al., 2021). In such cases, neural networks learn representations that capture spurious features and make predictions that employ them. Many approaches have been proposed for learning representations that do not capture spurious features (Muandet et al., 2013; Sun & Saenko, 2016; Ganin et al., 2016; Wang et al., 2019b;a; Li et al., 2018; Arjovsky et al., 2019; Zhao et al., 2020; Lu et al., 2022). Some methods are tailored against specific spurious correlations (e.g., texture); some require specifying a categorical spurious feature, while others require data collected from multiple labeled environments. Nevertheless, to our best knowledge, none of such representation learning methods consistently outperform standard empirical risk minimization (Gulrajani & Lopez-Paz, 2021; Koh et al., 2021). This is partly because spurious features are often easier to learn and get learned early in training (Shah et al., 2020; Nam et al., 2020; Hermann & Lampinen, 2020; Pezeshki et al., 2021).

Besides the unsatisfactory results, the approach mentioned above also goes against one of the main techniques of deep learning – using pretrained representations instead of learning from scratch. Recently, a few works indicated a large potential in fixing pretrained representations and focusing on training a linear classifier on top of it that does not rely on spurious correlations. In particular, Galstyan et al. (2022) find that a significant contribution to the out-of-domain generalization error comes from the classification head and call for designing better methods of training the classification head. Menon et al. (2021) propose to retrain the classification head on training data with down-sampled majority groups. Kirichenko et al. (2023); Izmailov et al. (2022); and Shi et al. (2023) find that after training on data with spurious correlations, keeping the representations fixed and retraining the classification head on small unbiased data gives state-of-the-art results. When no information about spurious features is available, Mehta et al. (2022) show that one can still get good results by using embeddings from a *large* pretrained vision model. Interestingly, representations learned by a

[1] YerevaNN [2] Yerevan State University [3] University of Southern California [4] Meta AI Research. Correspondence to: Hrayr Harutyunyan <hrayrhar@usc.edu>.

Published at the ICML 2023 Workshop on Spurious Correlations, Invariance, and Stability. Baltimore, Maryland, USA. Copyright 2023 by the author(s).

vision transformer (Dosovitskiy et al., 2021) seem to lead to more robust classification heads (Ghosal et al., 2022). Overall, these findings indicate that more research is needed to understand better how spurious features are represented and design better methods of training classification heads on representations that capture spurious features.

We consider the Waterbirds dataset (Sagawa* et al., 2020), which is landbird vs waterbird image classification task where the background is spuriously correlated with the label. Namely, most landbird images have land in their background, while most waterbird images have water in their background. We consider fixed pretrained representations learned through supervised or self-supervised learning. We investigate whether one can remove the spurious features from the representations in two settings. In the former (and more prevalent setting), one has access to the value of the binary spurious feature. In the latter, we also have access to per-example image masks indicating which parts of images correspond the spurious feature.

Interestingly, even with full knowledge of the spurious feature, it is not straightforward to remove it. While we find that representations are axis-aligned to a certain degree, the extent of alignment is not enough to remove spurious features by removing individual representation coordinates. Since both the spurious feature and the label can be predicted well from the representation with a linear layer, we hypothesize that the entanglement of core and spurious features is linear and can be reversed with a linear transformation. For this we propose a linear autoencoder to split the representation into three parts corresponding to the class label, the spurious feature, and other features not related to the former two but required for reconstruction. Importantly, in contrast to existing approaches, we do not enforce independence of the first and second parts on a biased training set. Instead, we enforce independence on an upsampled variant of the training set.

We find that a linear classifier trained on the core features of the encoding performs better than the standard approach but does not reach the performance of a classifier trained on an unbiased set. We demonstrate that this gap can be closed by performing additional feature selection within the core features.

2. Experimental Design

Waterbirds dataset. We consider the Waterbirds dataset (Sagawa et al., 2020), which is a benchmark dataset designed to measure the effect of debiasing spurious correlations. The dataset consists of bird photographs from the CUB (Welinder et al., 2010) dataset combined with image backgrounds from the Places (Zhou et al., 2018) dataset. It has 4,795 training examples, 1,199 val examples, and 5,794 testing examples. The task is to classify birds as waterbirds or landbirds while ignoring the background.

Bird masks. For every picture of the training set we use Detic (Zhou et al., 2022) to segment the bird in the image. The binary mask of the bird in the picture is denoted by m. We visually evaluate the quality of the masks and they are close to ideal. The most common error is that in rare cases there are additional birds in the background which are included in the mask.

Pretrained representations. We use three pretrained models in our experiments: ImageNet-pretrained ResNet-50, SWAG-pretrained and ImageNet-finetuned RegNetY, and a self-supervised ViT-B/14 from DINOv2. These models produce d-dimensional representations for each, where $d = 2048$, $d = 7392$, $d = 786$ for the three models respectively. Mehta et al. (2022) shows that the better models produce better worst group accuracy (WGA). Throughout this paper we use z to denote representations.

Attributing neurons. Whether or not individual neurons can be attributed to specific image regions is not clear. There is some evidence against it and some evidence in support of it (Elhage et al., 2022). Certainly some layers and operations in deep learning give preference to the standard basis. These include element-wise activations functions, batch normalization, dropout, etc. We use Captum (Kokhlikyan et al., 2020) to find this attribution. Captum has implementations of several attribution algorithms. In our preliminary experiments, we found that the results with the Integrated Gradients method (Sundararajan et al., 2017) are good enough.

For a given image $x_j \in \mathbb{R}^{224\times224}$, with foreground mask $m_j \in \{0,1\}^{224\times224}$, and representation $z_j \in \mathbb{R}^d$, we use Captum to compute attribution heatmap $a_j^i \in \mathbb{R}^{224\times224}$ of the neuron i-th neuron over input pixels. We define spuriousness $s(x_j) \in \mathbb{R}^d$ the following way:

$$s_i(x_j) = \frac{\sum (m_j \odot |a_j^i|)}{\sum m_j} - \frac{\sum ((1-m_j) \odot |a_j^i|)}{\sum (1-m_j)}, \quad (1)$$

where the sums are over pixels. The first term is the average attribution on the foreground pixels and the second term is the average attribution on the background pixels. Note that the attribution scores given by Captum can be both negative and positive, indicating the direction of the impact on the individual neuron. We use absolute values of the attribution scores as we are only interested in the magnitude of the impact. The spuriousness of the specific neuron z_i is defined as the average spuriousness over the training set X:

$$s_i = \frac{1}{|X|}\sum_{j=1}^{|X|} s_i(x_j). \quad (2)$$

Linear models. We train a linear classifier on a frozen representation using the *scikit-learn* package. We use L-BFGS optimizer and disable regularization. We report overall accuracy and worst group accuracy on the test set, and mostly focus on optimizing the latter. We note that SGD-based methods to learn a linear classifier can discover drastically different solutions, including solutions with higher worst group accuracy. This is especially true when early stopping is used. All linear models used in this paper use L-BFGS, and the analysis of SGD-based optimization is left for future work.

Upper bound. We compute an upper bound on worst group accuracy of L-BFGS-trained linear models by splitting the *test set* into five equal parts of 1000 samples, and perform five-fold cross-validation. This way we ensure that each of the five models is trained on a subset of the same size as the training set, but with equal number of images with water backgrounds and land backgrounds. The average of worst group accuracies of the five models is reported as an upper bound.

Statistical significance. For selected experiments we performed bootstrapping on the training set to estimate the variability of the models with respect to changes in the training set. We resampled with repetitions five versions of the training set of the same size, and repeated the experiments on the five versions. We report the mean and standard deviation of the five metrics.

3. Identifying Spurious Features

We start our investigations by studying whether one can remove spurious features by removing individual neurons from the pretrained representations.

Removing features that look mostly to background improves worst group accuracy. We sort the d neurons according to s_i and consider keeping only the top N neurons for the linear models. These models are denoted by $\text{Captum}^N(z, m)$. Figure 1 shows the results. In case of ResNet-50, a linear model on the top $N = 50$ neurons significantly improves the baseline (see also Table 1). The improvement is seen with up to $N = 260$ neurons. This confirms the hypothesis that there are many spurious features that harm the worst group accuracy of linear models, and s_i can be used as a measure to identify them. A similar effect is observed with RegNetY, there is improvement for at least up to $N = 1000$.

We could not detect this phenomenon in case of DINOv2. Keeping top neurons in terms of s_i worsens the metrics. Most likely this means that the individual neurons are neither pure spurious nor pure non-spurious. A supporting evidence is that the variance of s_i scores across the neurons is smaller than in case of ResNet-50.

Pretrained representations are relatively axis-aligned. To verify whether the directions in the representation space responsible for spurious features are aligned with the axes, we sample a random rotation matrix, apply it to the representations z, and recalculate s_i for them. Then we pick the top neurons from the rotated space and train new linear models. These models are denoted by $\text{Captum}^N(\text{Rot}(z), m)$. As seen in Figure 1, the scores are significantly worse in ResNet-50 (for $N \leq 250$) and RegNetY (for $N \leq 150$). This implies that the spuriousness directions are aligned with axes for these two representations. The difference is much smaller in DINOv2, which is expected, as there were no distinctive spurious neurons even before the rotation.

4. Disentangling Spurious and Core Features

Designing a group-aware autoencoder. As seen in the previous section, there exist spurious coordinates in ResNet-50 and RegNetY representations. We also showed that distinctive spuriousness is lost when we apply a random rotation matrix. This raises a question whether there exists another linear transformation that will make spurious features even more axis aligned, i.e. there will be new neurons that more specifically capture the spurious features. In other words, we are looking for ways to disentangle spurious and core features with a linear transformation.

We design a simple autoencoder where the linear encoder maps input z to three vectors: z_y, z_c and z_n. We force z_y and z_c to contain information about the label and the background, respectively. We do this by adding another linear layer on top of z_y and z_c that predict $\hat{y} = W_y z_y$ and $\hat{c} = W_c z_c$ which are supervised by the corresponding signals. The linear decoder takes the concatenation of z_y, z_c and z_n and reconstructs $\hat{z}$ which should be close to the original z. We assume that z_n will store the rest of the information in z that is not relevant for predicting either the label or the background. Note that we need group-level information to train this autoencoder, but we do not need the masks of the birds. Following (Jaiswal et al., 2019), we add an additional regularization term that minimizes mutual information between z_y and z_c. The final loss function is the following:

$$L = \text{ce}(\hat{y}, y) + \text{ce}(\hat{c}, c) + 10\|\hat{z} - z\| + 50\,\text{HSIC}(z_y, z_c), \quad (3)$$

where $\text{ce}(\cdot, \cdot)$ is the cross-entropy loss, and HSIC denotes the Hilbert-Schmidt Independence Criterion (Gretton et al., 2005).

We train the autoencoder on the *upsampled* version of the training set so that each group is represented equally. This also justifies the minimization of the mutual information between z_y and z_c, as they are correlated in the original training distribution. The sum of the dimensions of the three vectors z_y, z_c and z_n matches d for each backbone. The linear model trained on top of the concatenation of the three

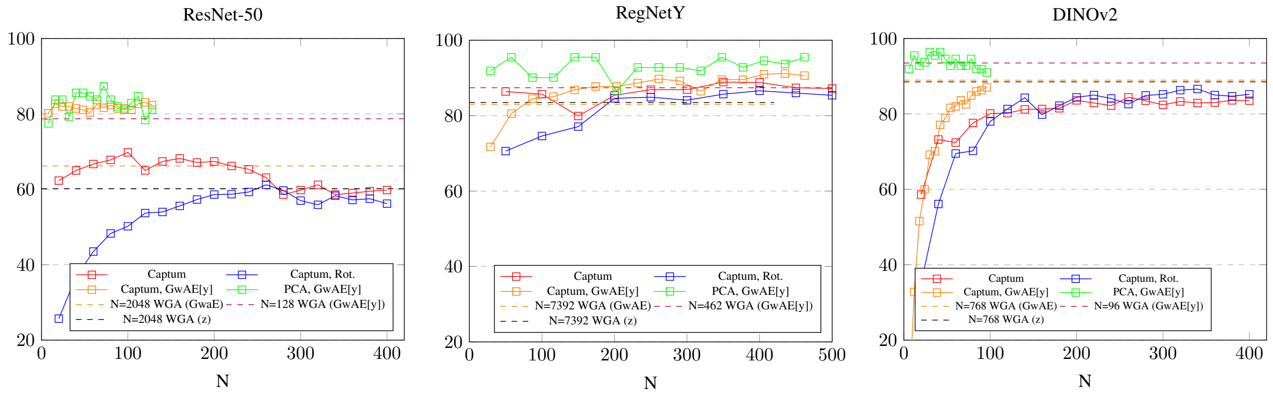

Figure 1: Worst group accuracy of linear models on subsets of N features selected using various methods.

vectors is denoted by GwAE(z, g), where GwAE denotes the linear encoder of the group-wise trained autoencoder, and g refers to the group information required for the training. GwAE$_y(z, g)$ denotes only the z_y part of the encoder's output. Note that the linear classifier trained on top of the autoencoder still belongs to the space of linear classifiers.

We expect the linear models trained on z and on the full GwAE(z, g) to perform similarly. We surprisingly see that this is not the case with ResNet-50. GwAE(z, g) is better by 6 percentage points. One explanation is that the autoencoder is not ideal, some information is lost by the encoder, and luckily the lost information contains some of the spurious features. We leave a deeper analysis for the future work.

Label-aware part has mostly good features. For all backbones we see that z_y gathers core features and the linear models trained on them have significantly better worst group accuracy. In case of ResNet-50 and DINOv2, the results are even better than the ones by Captum, which means that the autoencoder managed to isolate core features much better than it was possible by simply removing neurons in the original z. In case of RegNetY, we see that the linear model on z_y performs as good as many models trained on Captum-filtered neurons. This means that RegNetY features were already disentangled.

Furthermore, we apply Captum on top of the z_y neurons to see whether we can still identify and remove spurious features in z_y (the orange plot in Figure 1). This was successful only in case of ResNet-50.

PCA helps. To separate the core features, the linear encoder needs to shrink some directions (for example those corresponding to spurious features). For this reason, one can hypothesize that most of the variance in z_y will be along the core features. This motivates applying principal component analysis (PCA) on z_y to further remove non-core features. Unlike Captum, PCA does not require additional information from the data. We find that, indeed, training linear models on the N principal components of z_y still improves the worst group accuracy for ResNet and RegNetY. We note that applying PCA directly on z does not help to identify core features: the principal components of the original space usually contain spurious features.

Table 1: Results with ResNet-50.

Method	Accuracy	WGA
Standard training on z	83.1±1.2	60.2±1.8
PCA$^{20}(z)$	81.7±0.8	51.3±2.0
Captum$^{100}(z, m)$	88.0±0.3	67.7±1.3
Captum$^{300}(\text{Rot}(z), m)$	81.4±1.1	55.8±2.7
GwAE$(z, g)[y]$	91.2±0.3	78.7±0.3
PCA^{inf}(GwAE$(z, g)[y]$)	91.2±0.3	78.7±0.3
PCA^{20}(GwAE$(z, g)[y]$)	93.9±0.2	81.7±1.1
Captum^{20}(GwAE$(z, g), m$)	93.5±0.1	81.0±0.6
Captum^{20}(GwAE$(z, g)[y], m$)	93.5±0.1	81.0±0.6
Upper bound on z	92.3±0.7	82.9±3.9
Upper bound on GwAE(z, g)	92.6±1.1	81.1±5.0

5. Conclusion

With carefully designed experiments we have shown that pretrained image representations contain spurious features that can be identified and removed to improve worst group accuracy of the linear models. In most representations it is also possible to disentangle spurious features and further improve the performance. In future work we plan experiments with more backbones and datasets to see how well these findings generalize to other settings. We also note that it is possible to find linear models with better worst group accuracy if we use stochastic gradient descent with early stopping. This analysis is also left for future work.

References

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

Beery, S., Van Horn, G., and Perona, P. Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 456–473, 2018.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Elhage, N., Hume, T., Olsson, C., Schiefer, N., Henighan, T., Kravec, S., Hatfield-Dodds, Z., Lasenby, R., Drain, D., Chen, C., Grosse, R., McCandlish, S., Kaplan, J., Amodei, D., Wattenberg, M., and Olah, C. Toy models of superposition. *Transformer Circuits Thread*, 2022.

Galstyan, T., Harutyunyan, H., Khachatrian, H., Steeg, G. V., and Galstyan, A. Failure modes of domain generalization algorithms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19077–19086, 2022.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., and Brendel, W. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019.

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

Ghosal, S. S., Ming, Y., and Li, Y. Are vision transformers robust to spurious correlations? *arXiv preprint arXiv:2203.09125*, 2022.

Gretton, A., Bousquet, O., Smola, A., and Schölkopf, B. Measuring statistical dependence with hilbert-schmidt norms. In *Algorithmic Learning Theory: 16th International Conference, ALT 2005, Singapore, October 8-11, 2005. Proceedings 16*, pp. 63–77. Springer, 2005.

Gulrajani, I. and Lopez-Paz, D. In search of lost domain generalization. In *International Conference on Learning Representations*, 2021.

Gururangan, S., Swayamdipta, S., Levy, O., Schwartz, R., Bowman, S., and Smith, N. A. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 107–112, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2017.

Hermann, K. and Lampinen, A. What shapes feature representations? exploring datasets, architectures, and training. *Advances in Neural Information Processing Systems*, 33: 9995–10006, 2020.

Izmailov, P., Kirichenko, P., Gruver, N., and Wilson, A. G. On feature learning in the presence of spurious correlations. *Advances in Neural Information Processing Systems*, 35:38516–38532, 2022.

Jaiswal, A., Brekelmans, R., Moyer, D., Steeg, G. V., AbdAlmageed, W., and Natarajan, P. Discovery and separation of features for invariant representation learning. *arXiv preprint arXiv:1912.00646*, 2019.

Kirichenko, P., Izmailov, P., and Wilson, A. G. Last layer re-training is sufficient for robustness to spurious correlations. In *The Eleventh International Conference on Learning Representations*, 2023.

Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Gao, I., et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pp. 5637–5664. PMLR, 2021.

Kokhlikyan, N., Miglani, V., Martin, M., Wang, E., Alsallakh, B., Reynolds, J., Melnikov, A., Kliushkina, N., Araya, C., Yan, S., and Reblitz-Richardson, O. Captum: A unified and generic model interpretability library for pytorch, 2020.

Li, Y., Tian, X., Gong, M., Liu, Y., Liu, T., Zhang, K., and Tao, D. Deep domain generalization via conditional invariant adversarial networks. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 624–639, 2018.

Lu, C., Wu, Y., Hernández-Lobato, J. M., and Schölkopf, B. Invariant causal representation learning for out-of-distribution generalization. In *International Conference on Learning Representations*, 2022.

McCoy, T., Pavlick, E., and Linzen, T. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3428–3448, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1334.

Mehta, R., Albiero, V., Chen, L., Evtimov, I., Glaser, T., Li, Z., and Hassner, T. You only need a good embeddings extractor to fix spurious correlations. *arXiv preprint arXiv:2212.06254*, 2022.

Menon, A. K., Rawat, A. S., and Kumar, S. Overparameterisation and worst-case generalisation: friend or foe? In *International Conference on Learning Representations*, 2021.

Muandet, K., Balduzzi, D., and Schölkopf, B. Domain generalization via invariant feature representation. In *International conference on machine learning*, pp. 10–18. PMLR, 2013.

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.

Pezeshki, M., Kaba, O., Bengio, Y., Courville, A. C., Precup, D., and Lajoie, G. Gradient starvation: A learning proclivity in neural networks. *Advances in Neural Information Processing Systems*, 34:1256–1272, 2021.

Ribeiro, M. T., Singh, S., and Guestrin, C. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144, 2016.

Sagawa*, S., Koh*, P. W., Hashimoto, T. B., and Liang, P. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2020.

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, 2020.

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.

Shi, Y., Daunhawer, I., Vogt, J. E., Torr, P., and Sanyal, A. How robust is unsupervised representation learning to distribution shift? In *The Eleventh International Conference on Learning Representations*, 2023.

Sun, B. and Saenko, K. Deep coral: Correlation alignment for deep domain adaptation. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14*, pp. 443–450. Springer, 2016.

Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks. In *International conference on machine learning*, pp. 3319–3328. PMLR, 2017.

Torralba, A. and Efros, A. A. Unbiased look at dataset bias. In *CVPR 2011*, pp. 1521–1528. IEEE, 2011.

Wang, H., Ge, S., Lipton, Z., and Xing, E. P. Learning robust global representations by penalizing local predictive power. *Advances in Neural Information Processing Systems*, 32, 2019a.

Wang, H., He, Z., and Xing, E. P. Learning robust representations by projecting superficial statistics out. In *International Conference on Learning Representations*, 2019b.

Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., and Perona, P. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

Xiao, K. Y., Engstrom, L., Ilyas, A., and Madry, A. Noise or signal: The role of image backgrounds in object recognition. In *International Conference on Learning Representations*, 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.

Zhao, S., Gong, M., Liu, T., Fu, H., and Tao, D. Domain generalization via entropy regularization. *Advances in Neural Information Processing Systems*, 33:16096–16107, 2020.

Zhou, B., Lapedriza, À., Khosla, A., Oliva, A., and Torralba, A. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40:1452–1464, 2018.

Zhou, X., Girdhar, R., Joulin, A., Krähenbühl, P., and Misra, I. Detecting twenty-thousand classes using image-level supervision. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IX*, pp. 350–368. Springer, 2022.

A. Appendix

In Tables 2 and 3 we show the detailed results for RegNetY and DINOv2.

Table 2: Results with RegNetY.

	Accuracy	**WGA**
z	95.2±0.3	83.5±2.2
PCA$^{20}(z)$	90.7±0.5	74.6±1.7
Captum$^{350}(z, m)$	94.9±0.3	88.0±1.7
Captum^{350}(Rot$(z), m)$	93.5±0.8	85.6±1.1
GwAE$(z, g)[y]$	96.4±0.3	87.4±0.9
PCA^{inf}(GwAE$(z, g)[y])$	95.9±0.3	89.7±0.9
PCA^{20}(GwAE$(z, g)[y])$	96.5±0.1	90.4±1.7
Captum^{20}(GwAE$(z, g), m)$	96.3±0.2	88.9±0.3
Captum^{20}(GwAE$(z, g)[y], m)$	96.3±0.2	88.9±0.3
Upper bound on z	98.4±0.3	94.1±2.0
Upper bound on GwAE(z, g)	98.3±0.3	92.8±2.3

Table 3: Results with DINOv2.

	Accuracy	**WGA**
z	95.9±0.3	88.5±0.9
PCA$^{20}(z)$	93.7±0.4	80.6±1.4
Captum$^{100}(z, m)$	92.1±0.6	79.8±0.6
Captum^{100}(Rot$(z), m)$	92.1±0.1	76.6±2.4
Captum$^{700}(z, m)$	96.3±0.3	88.6±0.6
Captum^{700}(Rot$(z), m)$	96.3±0.4	89.2±1.9
GwAE$(z, g)[y]$	96.6±0.3	93.5±0.4
PCA^{inf}(GwAE$(z, g)[y])$	96.8±0.2	93.0±0.6
PCA^{20}(GwAE$(z, g)[y])$	97.4±0.2	94.0±0.8
Captum^{100}(GwAE$(z, g), m)$	96.6±0.4	90.6±0.8
Captum^{20}(GwAE$(z, g), m)$	94.1±0.2	83.8±1.2
Captum^{20}(GwAE$(z, g)[y], m)$	97.0±0.1	92.7±0.5
Upper bound on z	98.3±0.1	94.6±1.1
Upper bound on GwAE(z, g)	98.0±0.2	93.9±1.9